A Bibliography of Aspect-Oriented Software Development Version 1.23

Robert E. Filman
Research Institute for Advanced Computer Science
NASA Ames Research Center
Moffett Field, California

July 8, 2004

A bibliography of the literature related to Aspect-Oriented Programming.

My thanks to Mario Südholt, Karl Lieberherr, Sacher Dominik, Guenter Kniesel, Bart de Win, Lee Carver, Curtis Clifton and Mira Mezini for their contributions.

This is meant to be an evolving document. Please send additions, corrections, comments and disagreements to the author at *rfilman@mail.arc.nasa.gov*. There's more detail in the bibtex version of this file (see, for example, http://www.aosd.net/technology/aosd-bibliography.bib).

References

[1] Rickard A. Åberg, Julia L. Lawall, Mario Südholt, and Gilles Muller. Evolving an OS kernel using temporal logic and aspect-oriented programming. In Coady et al. [200].

[2] Franz Achermann. Language support for feature mixing. In ICSE-AOP00 [427].

[3] Franz Achermann. *Forms, Agents and Channels - Defining Composition Abstraction with Style*. PhD thesis, University of Berne, January 2002.

[4] Franz Achermann and Oscar Nierstrasz. Explicit Namespaces. In Jürg Gutknecht and Wolfgang Weck, editors, *Modular Programming Languages*, volume 1897 of *LNCS*, pages 77–89. Springer-Verlag, September 2000.

[5] Franz Achermann and Oscar Nierstrasz. Applications = Components + Scripts – A Tour of Piccola. In Mehmet Akşit, editor, *Software Architectures and Component Technology*, pages 261–292. Kluwer, 2001.

[6] M. Akşit. The analysis and design of distributed systems. In *Addendum to the Proc. on Object-Oriented Programming Systems, Languages and Applications*, page 223, 1992.

[7] M. Akşit. Composition and separation of concerns in the object-oriented model. *ACM Computing Surveys*, 28A(4), 1996.

[8] M. Akşit, editor. *Software Architectures and Component Technology: The State of the Art in Research and Practice*. Kluwer Academic Publishers, 2001.

[9] M. Akşit and L. Bergmans. Composing multiple-client–multiple-server synchronizations. In *Proc. IEEE Joint Workshop on Parallel and Distributed Systems*, pages 269–282, April 1997.

[10] M. Akşit and L. Bergmans. Examples of reusing synchronization code in aspect-oriented programming using composition-filters. In *Proc. 5th. Maghrebian Conf. Software Engineering and Artificial Intelligence (MCSEAI'98)*, pages 257–272, December 1998.

[11] M. Akşit and L. Bergmans. Guidelines for identifying obstacles when composing distributed systems from components. In Akşit [8], pages 29–56.

[12] M. Akşit, L. Bergmans, and S. Vural. An object-oriented language-database integration model: The composition-filters approach. In O. Lehrmann Madsen, editor, *Proc. 7th European Conf. Object-Oriented Programming*, pages 372–395. Springer-Verlag Lecture Notes in Computer Science, 1992.

[13] M. Akşit and J. Bosch. Issues in object-oriented real time language design. In W. Halang and A. Stoyenko, editors, *Real Time Computing*, pages 510–511. Springer Verlag Nato ASI Series, 1992.

[14] M. Akşit, J. Bosch, W. v. d. Sterren, and L. Bergmans. Real-time specification inheritance anomalies and real-time filters. In Tokoro and Pareschi [829], pages 386–407.

[15] M. Akşit and B. Tekinerdoğan. Component composability issues in object-oriented programming. *XOOTIC Magazine*, 5(2):15–20, December 1997.

[16] M. Akşit and B. Tekinerdoğan. Reusing and composing components: Problems and solutions. In *Proc. Smalltalk und Java in Industrie und Ausbildung 97 (STJA 97)*, September 1997.

[17] M. Akşit and B. Tekinerdoğan. Aspect-oriented programming using composition filters. In S. Demeyer and J. Bosch, editors, *Object-Oriented Technology, ECOOP'98 Workshop Reader*, page 435. Springer Verlag, July 1998.

[18] M. Akşit and B. Tekinerdoğan. Solving the modeling problems of object-oriented languages by composing multiple aspects using composition filters. In ECOOP-AOP98 [277].

[19] M. Akşit, B. Tekinerdoğan, and L. Bergmans. Achieving adaptability through separation and composition of concerns. In M. Muhlhauser, editor, *Special Issues in Object-Oriented Programming*, pages 12–23. dpunkt verlag, 1996.

[20] M. Akşit, K. Wakita, J. Bosch, L. Bergmans, and A. Yonezawa. Abstracting object-interactions using composition-filters. In R. Guerraoui, O. Nierstrasz, and M. Riveill, editors, *Object-Based Distributed Processing*, pages 152–184. Springer-Verlag Lecture Notes in Computer Science, 1993.

[21] Mehmet Akşit. Issues in aspect-oriented programming. In ECOOP-AOP97 [276].

[22] Mehmet Akşit, editor. *Proc. 2nd Int' Conf. on Aspect-Oriented Software Development (AOSD-2003)*. ACM Press, March 2003.

[23] Mehmet Akşit and Ziéd Choukair, editors. *Proc. 2nd Int'l Workshop on Aspect Oriented Programming for Distributed Computing Systems (ICDCS-2002), Vol. 2*, July 2002.

[24] Mehmet Akşit and Mira Mezini, editors. *Net.Object Days 2002*, October 2002.

[25] Mehmet Akşit, Bedir Tekinerdoğan, and Lodewijk Bergmans. The six concerns for separation of concerns. In ECOOP-AOP01 [275].

[26] Faisal Akkai, Atef Bader, and Tzilla Elrad. Dynamic weaving for building reconfigurable software systems. In OOPSLA-AOP01 [629].

[27] Omar Aldawud, Atef Bader, and Tzilla Elrad. Weaving with statecharts. In AOSD-UML02 [52].

[28] Omar Aldawud, Tzilla Elrad, and Atef Bader. A UML profile for aspect oriented modeling. In OOPSLA-AOP01 [629].

[29] Omar Aldawud, Tzilla Elrad, and Atef Bader. UML profile for aspect-oriented software development. In Aldawud et al. [30].

[30] Omar Aldawud, Mohamed Kandé, Grady Booch, Bill Harrison, and Dominik Stein, editors. *Third International Workshop on Aspect Oriented Modeling*, March 2003.

[31] Omar Aldawud, Mohamed Kandé, Grady Booch, Bill Harrison, Dominik Stein, Jeff Gray, Siobhán Clarke, Aida Zakaria Santeon, Peri Tarr, and Faisal Akkawi, editors. *The 4th AOSD Modeling With UML Workshop*, October 2003.

[32] Jonathan Aldrich. Challenge problems for separation of concerns. In OOPSLA-AOP00 [628].

[33] Roger T. Alexander and James M. Bieman. Challenges of aspect-oriented technology. In *Workshop on Software Quality, 24th Int'l Conf. Software Engineering*, May 2002.

[34] Vander Alves and Paulo Borba. A design pattern for distributed applications. In Adriano Souza et al., editor, *XIV Brazilian Symposium on Software Engineering-Minicourses and Tutorials*, pages 191–219, oct 2000.

[35] Vander Alves and Paulo Borba. Distributed adapters pattern: A design pattern for object-oriented distributed applications. In *First Latin American Conference on Pattern Languages Programming, SugarLoafPLoP 2001*, oct 2001. Published in UERJ Magazine: Special Issue on Software Patterns, June 2002, pages 132-142.

[36] Vander Alves and Paulo Borba. An implementation method for distributed object-oriented applications. In *Proceedings of XV Brazilian Symposium on Software Engineering, SBES 2001*, pages 161–176, oct 2001.

[37] Vander Alves, Ayla Dantas, and Paulo Borba. Aop-driven variability in product lines of pervasive computing applications. In Matthias Riebisch, Jim Coplien, and Detlef Streitferdt, editors, *Modeling Variability for Object-Oriented Product Lines*, July 2003.

[38] Davide Ancona, Giovanni Lagorio, and Elena Zucca. Jam—a smooth extension of Java with mixins. In Bertino [105], pages 154–178.

[39] Ken Anderson. An example of using collaborator and adapters to reuse a synchronization pattern. In OOPSLA-AOP00 [628].

[40] L. Andrade and J. L. Fiadeiro. An architectural approach to auto-adaptive systems. In Akşit and Choukair [23].

[41] Luis Andrade, José Luiz Fiadeiro, João Gouveia, and Georgios Koutsoukos. Separating computation, coordination and configuration. *Journal of Software Maintenance and Evolution: Research and Practice*, 14(5):353–369, 2002.

[42] James Andrews. Using process algebra as a foundation for programming by separation of concerns. In ICSE-AOP01 [428].

[43] James H. Andrews. Process-algebraic foundations of aspect-oriented programming. In Yonezawa and Matsuoka [885], pages 187–209.

[44] James H. Andrews. Using process algebra as a foundation for programming by separation of concerns. In ECOOP-AOP01 [275].

[45] M. Antunes, H. Miranda, A. Rito Silva, L. Rodrigues, and J. Martins. Separating replication from distributed communication: Problems and solutions. In Choukair [165], pages 103–110.

[46] Jo ao Paulo Barros and Luis Gomes. Towards the support for crosscutting concerns in activity diagrams: a graphical approach. In Aldawud et al. [31].

[47] *Workshop on Identifying, Separating and Verifying Concerns in the Design (AOSD-2002)*, March 2002.

[48] *Workshop on Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design (AOSD-2002)*, March 2002.

[49] *FOAL 2002: Foundations of Aspect-Oriented Languages (AOSD-2002)*, March 2002.

[50] *First AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (AOSD-2002)*, March 2002.

[51] *Modular Representation and Interpretation of Concerns in XML (AOSD-2002)*, March 2002.

[52] *Workshop on Aspect-Oriented Modeling with UML (AOSD-2002)*, March 2002.

[53] J. Araujo and P. Coutinho. Identifying aspectual use cases using a viewpoint-oriented requirements method. In Araújo et al. [54].

[54] João Araújo, Awais Rashid, Bedir Tekinerdogan, Ana Moreira, and Paul Clements, editors. *Early Aspects 2003: Aspect-Oriented Requirements Engineering and Architecture Design*, March 2003.

[55] João Araújo, Ana Moreira, Isabel Brito, and Awais Rashid. Aspect-oriented requirements with uml. In Kandé et al. [443].

[56] Ali Arsanjani, Brent Hailpern, Joanne Martin, and Peri Tarr. Web services: Promises and compromises. *Queue*, 1(1):48–58, 2003.

[57] Uwe Aßmann. A component model for invasive composition. In ECOOP-AOP00 [274].

[58] Uwe Aßmann and Andreas Ludwig. Aspect weaving by graph rewriting. In U. W. Eisenecker and K. Czarnecki, editors, *Generative Component-based Software Engineering (GCSE)*, October 1999.

[59] Colin Atkinson and Thomas Kühne. Separation of concerns through stratified architectures. In ECOOP-AOP00 [274].

[60] S. Aussmann and M. Haupt. Axon—dynamic AOP through runtime inspection and monitoring. In Filman et al. [296].

[61] Enis Avdicaušević, Marjan Mernik, Mitja Lenic, and Viljem Zumer. Experimental aspect-oriented language - aspectcool. In *Proceedings of the 17th symposium on Proceedings of the 2002 ACM symposium on applied computing*, pages 943–947. ACM Press, 2002.

[62] Boris Bachmendo, Stefan Hanenberg, Stephan Herrmann, and Günter Kniesel, editors. *3rd Workshop on Aspect-Oriented Software Development (AOSD-GI) of the SIG Object-Oriented Software Development, German Informatics Society*, March 2003.

[63] Boris Bachmendo and Rainer Unland. Aspect-based workflow evolution. In Rashid [706].

[64] Atef Bader and Tzilla Elrad. The adaptive arena: Language constructs and architectural abstractions for concurrent object-oriented systems. In *ICPADS 98*, 1998.

[65] Atef Bader and Tzilla Elrad. Framework and design pattern for concurrent passive objects. In *Proc. IASTED/SE 98*, 1998.

[66] Jason Baker and Wilson Hsieh. Runtime aspect weaving through metaprogramming. In Kiczales [473], pages 86–98.

[67] Shubhanan Bakre and Tzilla Elrad. Linguistic issues for developing aspect-rich high-performance adaptable systems. In Bergmans et al. [98].

[68] Elisa Baniassad, Gail Murphy, Christa Schwanninger, and Michael Kircher. Managing crosscutting concerns during software evolution tasks: An inquisitive study. In Kiczales [473], pages 120–126.

[69] Elisa L. A. Baniassad, Gail C. Murphy, and Christa Schwanninger. Determining the "why" of concerns. In ECOOP-AOP01 [275].

[70] Elisa L. A. Baniassad, Gail C. Murphy, and Christa Schwanninger. Determining the "why" of concerns. In ICSE-AOP01 [428].

[71] Elisa L. A. Baniassad, Gail C. Murphy, Christa Schwanninger, and Michael Kircher. Where are programmers faced with concerns? In OOPSLA-AOP00 [628].

[72] D. Bardou. Roles, subjects and aspects: How do they relate? In ECOOP-AOP98 [277].

[73] Daniel Bardou. Experimenting split-object in Java and AspectJ. In Bachmendo et al. [62].

[74] Luciano Porto Barreto, Rèmi Douence, Gilles Muller, and Mario Südholt. Programming OS schedulers with domain-specific languages and aspects: New approaches for OS kernel engineering. In AOSD-PAT02 [50].

[75] João Paulo Barros and Luís Gomes. Activities as behaviour aspects. In Kandé et al. [443].

[76] Mark Basch and Arturo Sanchez. Incorporating aspects into the UML. In Aldawud et al. [30].

[77] Paul G. Bassett. *Framing Software Reuse: Lessons From the Real World*. Prentice Hall PTR, Englewood Cliffs, New Jersey, 1996.

[78] Adam Batenin and Eamonn O'Neill. Towards unanticipated composition of concerns in hyperspaces. In AOSD-AOD02 [47].

[79] Don Batory. Refinements and separation of concerns. In ICSE-AOP00 [427].

[80] Don Batory, Clay Johnson, Bob MacDonald, and Dale von Heeder. Achieving extensibility through product-lines and domain-specific languages: A case study. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(2):191–214, 2002.

[81] Don Batory, Robert E. Lopez-Herrejon, and Jean-Philippe Martin. Generating product-lines of product-families. In *Proc. 17th IEEE Int'l Conf. on Automated Software Engineering*, pages 81–92, September 2002.

[82] Joachim Bayer. Towards engineering product lines using concerns. In ICSE-AOP00 [427].

[83] Christian Becker. Quality of service and O.O. oriented middleware multiple concerns and their separation. In Choukair [165], pages 117–126.

[84] Christian Becker and Kurt Geihs. Quality of service - aspects of distributed programs. In ICSE-AOP98 [429].

[85] U. Becker. D2AL: A design-based aspect language for distribution control. In ECOOP-AOP98 [277].

[86] Georg Beier and Markus Kern. Aspects in uml models from a code generation perspective. In Kandé et al. [443].

[87] L. Berger, A. M. Dery, and M. Fornarino. Interactions between objects: An aspect of object-oriented languages. In ICSE-AOP98 [429].

[88] L. Berger, A.M. Dery, and M. Fornarino. Interactions between objects: An aspect of object-oriented languages. In ECOOP-AOP98 [277].

[89] L. Bergmans. *Composing Concurrent Objects*. PhD thesis, University of Twente, 1994.

[90] L. Bergmans. The composition filters object model. Technical report, Dept. of Computer Science, University of Twente, 1994.

[91] L. Bergmans and M. Akşit. Reusability problems in object-oriented concurrent programs. In *Proc. ECOOP'92 Workshop Object-Based Concurrency and Reuse*, June 1992.

[92] L. Bergmans and M. Akşit. Composing synchronisation and real-time constraints. *Journal of Parallel and Distributed Computing*, 36:32–52, 1996.

[93] L. Bergmans and M. Akşit. Aspects and crosscutting in layered middleware systems. In *RM2000 Workshop in Reflective Middleware*, April 2000.

[94] L. Bergmans and M. Akşit. Composing crosscutting concerns using composition filters. *Comm. ACM*, 44(10):51–57, October 2001.

[95] L. Bergmans, M. Akşit, and J. Bosch. Composition filters: Extended expressiveness for oopls. In *OOPSLA'92 Workshop Object-Oriented Programming Languages: The Next Generation*, 1992.

[96] L. Bergmans, M. Akşit, and B. Tekinerdoğan. Aspect composition using composition filters. In Akşit [8], pages 357–382.

[97] Lodewijk Bergmans and Mehmet Akşit. Composing software from multiple concerns: A model and composition anomalies. In ICSE-AOP00 [427].

[98] Lodewijk Bergmans, Johan Brichau, Peri Tarr, and Erik Ernst, editors. *SPLAT: Software engineering Properties of Languages for Aspect Technologies*, March 2003.

[99] Lodewijk Bergmans, Bedir Tekinerdoğan, Maurice Glandrup, and Mehmet Akşit. On composing separated concerns: Composability and composition anomalies. In OOPSLA-AOP00 [628].

[100] Lodewijk M. J. Bergmans and Mehmet Akşit. Analyzing multi-dimensional programming in AOP and composition filters. In OOPSLA-AOP99 [631].

[101] Lodewijk M. J. Bergmans and Mehmet Akşit. How to deal with encapsulation in aspect-orientation. In OOPSLA-AOP01 [629].

[102] Lodewijk M.J. Bergmans. Towards detection of semantic conflicts between crosscutting concerns. In Hannemann et al. [377].

[103] Technologies Lodewijk Bergmans, Kris Gybels, Peri Tarr, and Erik Ernst, editors. *SPLAT: Software engineering Properties of Languages for Aspect*, March 2004.

[104] Paul Bergstein. *Managing the Evolution of Object-Oriented Systems*. PhD thesis, Northeastern University, 1994.

[105] E. Bertino, editor. *ECOOP 2000—Object-Oriented Programming: 14th European Conference, LNCS 1850*. Springer Verlag, June 2000.

[106] Antoine Beugnard. How to make aspect reusable, a proposition. In ECOOP-AOP99 [278].

[107] Jean Bezivin. Aspect-oriented modeling: Oxymoron or pleonasm? In Akşit and Choukair [23].

[108] Andrew P. Black and Mark P. Jones. Perspectives on software. In OOPSLA-AOP00 [628].

[109] Andrew P. Black and Jonathan Walpole. Aspects of information flow. In ECOOP-AOP00 [274].

[110] L. Blair and G. S. Blair. The impact of aspect-oriented programming on formal methods. In ECOOP-AOP98 [277].

[111] L. Blair and G. S. Blair. The impact of aspect-oriented programming on formal methods. Technical Report MPG-98-08, Lancester University, 1998.

[112] L. Blair, G. S. Blair, and A. Andersen. Separating function behaviour and performance contraints: Aspect-oriented specification. Technical Report MPG-98-07, Lancester University, 1998.

[113] L. Blair, G. S. Blair, R. Andersen, and T. Jones. Formal support for dynamic qos management in the development of open component-based distributed systems. *IEE Proceedings-Software*, 148(3):89–97, June 2001.

[114] Lynne Blair and Gordon Blair. A tool suite to support aspect-oriented specification. In ECOOP-AOP99 [278].

[115] Lynne Blair and Mattia Monga. Reasoning on AspectJ programmes. In Bachmendo et al. [62].

[116] G. Blank and G. Vayngrib. Aspects of Enterprise Java Beans. In ECOOP-AOP98 [277].

[117] M. Blay-Fornarino, A. M. Pinna-Dery, and M. Riveill. Configuring distributed applications. In Akşit and Choukair [23].

[118] Christoph Bockisch, Michael Haupt, Mira Mezini, and Klaus Ostermann. Virtual machine support for dynamic join points. In Lieberherr [520], pages 83–92.

[119] Ron Bodkin. Commercialization of aosd: The road ahead. In Bodkin et al. [120].

[120] Ron Bodkin, Adrian Colyer, Juri Memmert, and Arno Schmidmeier, editors. *AOSD Workshop on Commercialization of AOSD Technology*, March 2003.

[121] K. Böllert. Aspect-oriented programming case study: System management application. In ECOOP-AOP98 [277].

[122] Kai Böllert. On weaving aspects. In ECOOP-AOP99 [278].

[123] Jonas Bonér. What are the key issues for commercial AOP use—how does AspectWerkz address them? In Lieberherr [520], pages 5–6.

[124] J. Bosch and M. Akşit. Composition-filters based real-time programming. In *OOPSLA'92 Workshop on Evaluation of Object-Oriented Technology in Real-Time Systems: Past, Present & Future*, 1992.

[125] Jan Bosch, editor. *Generative and Component-Based Software Engineering, Third International Conference (GCSE 2001), LNCS 2186*. Springer-Verlag Lecture Notes in Computer Science, September 2001.

[126] Philippe Bouaziz and Lionel Seinturier. From software parameterization to software profiling. In Rashid [706].

[127] Noury Bouraqadi. Concern oriented programming using reflection. In OOPSLA-AOP00 [628].

[128] Noury M. N. Bouraqadi-Saâdanii and Thomas Ledoux. How to weave? In ECOOP-AOP01 [275].

[129] Laurent Boussard. Towards a pragmatic composition model of CORBA services based on AspectJ. In ECOOP-AOP00 [274].

[130] Alexandre M. Braga, Ricardo Darab, and Cecília M. F. Rubira. A meta-object protocol for secure composition of security mechanisms. In OOPSLA-AOP00 [628].

[131] Johan Brichau. Declarative composable aspects. In OOPSLA-AOP00 [628].

[132] Johan Brichau, Wolfgang De Meuter, and Kris De Volder. Jumping aspects. In ECOOP-AOP00 [274].

[133] I. Brito and A. Moreira. Towards a composition process for aspect-oriented requirements. In Araújo et al. [54].

[134] I. Brito, A. Moreira, and J. Araujo. A requirements model for quality attributes. In AOSD-EA02 [48].

[135] Alex Brodsky, Dima Brodsky, Ida Chan, Yvonne Coady, Stephan Gudmundson, Jody Pomkoski, and Joon Suan Ong. Coping with evolution: Aspects vs. aspirin. In OOPSLA-AOP01 [629].

[136] Adam Brown, Richard Cardone, Sean McDirmid, and Calvin Lin. Using mixins to build flexible widgets. In Kiczales [473], pages 76–85.

[137] David Bruce and Nick Exon. Alternatives to aspect-oriented programming? In Rashid [706].

[138] Jean-Michel Bruel, Jo ao Araújo, Ana Moreira, and Albert Royer. Using aspects to develop built-in tests for components. In Aldawud et al. [31].

[139] Eric Bruneton and Michel Riveill. Experiments with JavaPod, a platform designed for the adaptation of non-functional properties. In Yonezawa and Matsuoka [885], pages 52–72.

[140] Avi Bryant, Andrew Catton, Kris De Volder, and Gail Murphy. Explicit programming. In Kiczales [473], pages 10–18.

[141] Martin Büchi and Wolfgang Weck. Generic wrappers. In Bertino [105], pages 201–225.

[142] R. J. A. Buhr. A possible design notation for aspect-oriented programming. In ECOOP-AOP98 [277].

[143] G. Cabri, L. Leonardi, and F. Zambonelli. Separation of concerns in agent applications by roles. In Akşit and Choukair [23].

[144] Joäo Cachopo, António Menezes Leitäo, and António Rito-Silva. The tyranny of the file decomposition. In OOPSLA-AOP02 [630].

[145] V. Cahill. An overview of the Coyote project. In ECOOP-AOP98 [277].

[146] A. Cain, J. Schneider, D. Grant, and T. Chen. Runtime data analysis for Java programs. In Filman et al. [296].

[147] Sabine Canditt and Manfed Gunter. Aspect oriented logging in a real-world system. In AOSD-PAT02 [50].

[148] Adeline Capouillez, Pierre Crescenzo, and Philippe Lahire. Separation of concerns in OFL. In ECOOP-AOP01 [275].

[149] Lee Carver. Combining selector-guarded blocks. In ECOOP-AOP00 [274].

[150] Lee Carver. A practical hyperspace application: Lessons from the option-processing task. In ICSE-AOP00 [427].

[151] Lee Carver. Using brackets to corral jumping aspects. In OOPSLA-AOP00 [628].

[152] Lee Carver. Building a real-world application with aspect-oriented modules and hyper/j. Master's thesis, University of California, San Diego, June 2002.

[153] Lee Carver. Composition behaviors for application construction. In AOSD-AOD02 [47].

[154] Lee Carver. Simplified universe construction for Hyper/J composition. In OOPSLA-AOP02 [630].

[155] Lee Carver. Next steps for commercializing AOP. In Bodkin et al. [120].

[156] Lee Carver and William G. Griswold. Sorting out concerns. In OOPSLA-AOP99 [631].

[157] Christina Chavez, Alessandro Garcia, and Carlos Lucena. Some insights on the use of AspectJ and Hyper/J. In Rashid [706].

[158] Christina Chavez and Carlos Lucena. A metamodel for aspect-oriented modeling. In AOSD-UML02 [52].

[159] Marsha Chechik and Steve Easterbrook. Reasoning about compositions of concerns. In ICSE-AOP01 [428].

[160] S. Chiba, Y. Sato, and M. Tatsubori. Using hotswap for implementing dynamic aop systems. In Filman et al. [296].

[161] Shigeru Chiba. What are the best join points? In OOPSLA-AOP01 [629].

[162] Shigeru Chiba and Kiyoshi Nakagawa. Josh: An open aspectj-like language. In Lieberherr [520], pages 102–111.

[163] Ruzanna Chitchyan, Ian Sommerville, and Awais Rashid. An analysis of design approaches for crosscutting concerns. In AOSD-AOD02 [47].

[164] Ruzanna Chitchyan, Ian Sommerville, and Awais Rashid. A model for dynamic hyperspaces. In Bergmans et al. [98].

[165] Ziéd Choukair, editor. *Proc. Int'l Workshop on Distributed Dynamic Multiservice Architectures (ICDCS-2001), Vol. 2*, April 2001.

[166] Mark Chu-Carroll. Software configuration management as a mechanism for multidimensional separation of concerns. In ICSE-AOP00 [427].

[167] Mark C. Chu-Carroll. Separation of concerns: An organizational approach. In OOPSLA-AOP00 [628].

[168] Mark C. Chu-Carroll. Separation of concerns in software configuration management. In ICSE-AOP01 [428].

[169] Mark C. Chu-Carroll. Separation of concerns in software configuration management. In ECOOP-AOP01 [275].

[170] Mark C. Chu-Carroll. Supporting aspects in program storage. In OOPSLA-AOP02 [630].

[171] Mark C. Chu-Carroll, James Wright, and Annie T. T. Ying. Visual separation of concerns through multidimensional program storage. In Akşit [22], pages 188–197.

[172] Maria Agustina Cibran and Maja D'Hondt. Composable and reusable business rules using AspectJ. In Bergmans et al. [98].

[173] M. Cilia, M. Haupt, M. Mezini, and A. P. Buchmann. The convergence of AOP and active databases: Towards reactive middleware. In F. Pfenning and Y. Smaragdakis, editors, *Proceedings of 2nd International Conference on Generative Programming and Component Engineering (GPCE)*, volume 2830 of *LNCS*, pages 169–188. Springer-Verlag, 2003.

[174] Tony Clark, Andy Evans, and Stuart Kent. Aspect-oriented metamodelling. *The Computer Journal*, 46(5):566–577, September 2003.

[175] Siobhán Clarke. Composing design models: An extension to the UML. In *Proc. Third Int'l Conf. the Unified Modeling Language (UML), LNCS 1939*, pages 338–352. Springer-Verlag, October 2000.

[176] Siobhán Clarke. Designing reusable patterns of cross-cutting behaviour with composition patterns. In OOPSLA-AOP00 [628].

[177] Siobhán Clarke. Extending UML metamodel for design composition. In ICSE-AOP00 [427].

[178] Siobhán Clarke. *Composition of Object-Oriented Software Design Models*. PhD thesis, Dublin City University, January 2001.

[179] Siobhán Clarke. Extending standard UML with model composition semantics. *Science of Computer Programming*, to appear.

[180] Siobhán Clarke, William Harrison, Harold Ossher, and Peri Tarr. Designing for evolution with subjects. In *Workshop on Software Change and Evolution, ICSE*, 1999.

[181] Siobhán Clarke, William Harrison, Harold Ossher, and Peri Tarr. The dimensions of separating requirements concerns for the duration of the development lifecycle. In OOPSLA-AOP99 [631].

[182] Siobhán Clarke, William Harrison, Harold Ossher, and Peri Tarr. Separating concerns throughout the development lifecycle. In ECOOP-AOP99 [278].

[183] Siobhán Clarke, William Harrison, Harold Ossher, and Peri Tarr. Subject-oriented design: Towards improved alignment of requirements, design and code. In *Proc. Object-Oriented Programming, Systems, Languages and Applications (OOPSLA)*, pages 325–339, November 1999.

[184] Siobhán Clarke, William Harrison, Harold Ossher, and Peri Tarr. Subject-oriented design: Support for evolution from the design stage. In *Workshop on Software and Organisation Co-Evolution*, 1999.

[185] Siobhán Clarke, William Harrison, Harold Ossher, and Peri Tarr. Subject-oriented design: Towards improved alignment of requirements, design and code. Research Report RC 21300, IBM, April 1999.

[186] Siobhán Clarke and John Murphy. Developing a tool to support composition of the components in a large-scale development. In *OOPSLA Workshop on Object-Oriented Behavioural Semantics*, 1997.

[187] Siobhán Clarke and John Murphy. Developing a tool to support the application of aspect-oriented programming principles to the design phase. In ICSE-AOP98 [429].

[188] Siobhán Clarke and John Murphy. Verifying components under development at the design stage: A tool to support the composition of component design models. In *ICSE Workshop on Component-Based Software Engineering*, 1998.

[189] Siobhán Clarke, John Murphy, and Mark Roantree. Composition of UML design models: A tool to support the resolution of conflicts. In *Proc. Object Oriented Information Systems (OOIS)*, 1998.

[190] Siobhán Clarke and Robert Walker. Towards a standard design language for AOSD. In Kiczales [473], pages 113–119.

[191] Siobhán Clarke and Robert J. Walker. Composition patterns: An approach to designing reusable aspects. In *Proc. 23rd Int'l Conf. Software Engineering (ICSE)*, pages 5–14, May 2001.

[192] Siobhán Clarke and Robert J. Walker. Mapping composition patterns to AspectJ and Hyper/J. In ECOOP-AOP01 [275].

[193] Siobhán Clarke and Robert J. Walker. Mapping composition patterns to AspectJ and Hyper/J. In ICSE-AOP01 [428].

[194] Siobhán Clarke and Robert J Walker. Separating crosscutting concerns across the lifecycle: From composition patterns to AspectJ and Hyper/J. Technical Report TCD-CS-2001-15, Trinity College, Dublin, May 2001.

[195] Andy Clement, Adrian Colyer, and Mik Kersten. Aspect-oriented programming with ajdt. In Hannemann et al. [377].

[196] Pedro J. Clemente and Juan Hernández. Aspect component based software engineering. In Coady et al. [200].

[197] Curtis Clifton and Gary T. Leavens. Observers and assistants: A proposal for modular aspect-oriented reasoning. In AOSD-FOAL02 [49], pages 33–44.

[198] Curtis Clifton and Gary T. Leavens. Obliviousness, modular reasoning, and the behavioral subtyping analogy. In Bergmans et al. [98].

[199] Yvonne Coady, Alex Brodsky, Dima Brodsky, Jody Pomkoski, Stephan Gudmundson, Joon Suan Ong, and Gregor Kiczales. Can AOP support extensibility in client-server architectures? In ECOOP-AOP01 [275].

[200] Yvonne Coady, Eric Eide, and David H. Lorenz, editors. *The Second AOSD Workshop on Aspects, Components, and Patterns for Infrastructure Software (ACP4IS)*, March 2003.

[201] Yvonne Coady and Gregor Kiczales. Back to the future: A retroactive study of aspect evolution in operating system code. In Akşit [22], pages 50–59.

[202] Yvonne Coady, Gregor Kiczales, and Michael Feeley. Exploring an aspect-oriented approach to operating system code. In OOPSLA-AOP00 [628].

[203] Yvonne Coady, Gregor Kiczales, Mike Feeley, Norm Hutchinson, and Joon Suan Ong. Structuring operating system aspects: Using AOP to improve OS structure modularity. *Comm. ACM*, 44(10):79–82, October 2001.

[204] Yvonne Coady, Gregor Kiczales, Mike Feeley, Norm Hutchinson, and Joon Suan Ong. Structuring system aspects. In ICSE-AOP01 [428].

[205] Yvonne Coady, Gregor Kiczales, Mike Feeley, and Greg Smolyn. Using aspectc to improve the modularity of path-specific customization in operating system code. In *Proceedings of the 8th European Software Engineering Conference held jointly with 9th ACM SIGSOFT Symposium on Foundations of Software Engineering*, pages 88–98. ACM Press, 2001.

[206] Geoff A. Cohen. Recombing concerns: Experience with transformation. In OOPSLA-AOP99 [631].

[207] Geoff A. Cohen. A taxonomy of transformation. In OOPSLA-AOP00 [628].

[208] T. Colcombet and P. Fradet. Enforcing trace properties by program transformation. In *Proc. 27th ACM Symp. on Principles of Programming Languages*, pages 54–66, January 2000.

[209] Adrian Colyer. Towards widespread adoption of aosd. In Bodkin et al. [120].

[210] Adrian Colyer, Gordon Blair, and Awais Rashid. Managing complexity in middleware. In Coady et al. [200].

[211] Adrian Colyer and Andrew Clement. Large-scale aosd for middleware. In Lieberherr [520], pages 56–65.

[212] C. Constantinides and T. Skotiniotis. Reasoning about a classification of cross-cutting concerns in object-oriented systems. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[213] Constantinos Constantinides, Atef Bader, and Tzilla Elrad. An aspect-oriented design framework for concurrent systems. In ECOOP-AOP99 [278].

[214] Constantinos A. Constantinides. A case study on making the transition from functional to and fine-grained decomposition. In Hannemann et al. [377].

[215] Constantinos A. Constantinides, Atef Bader, and Tzilla Elrad. A framework to address a two-dimensional composition of concerns. In OOPSLA-AOP99 [631].

[216] Constantinos A. Constantinides, Atef Bader, and Tzilla Elrad. Separation of concerns in concurrent software systems. In ECOOP-AOP00 [274].

[217] Constantinos A. Constantinides, Atef Bader, Tzilla H. Elrad, P. Netinant, and Mohamed E. Fayad. Designing an aspect-oriented framework in an object-oriented environment. *ACM Computing Surveys*, 32(1es):41, 2000.

[218] Constantinos A. Constantinides and Tzilla Elrad. On the requirements for concurrent software architectures to support advanced separation of concerns. In OOPSLA-AOP00 [628].

[219] Constantinos A. Constantinides and Tzilla Elrad. Towards a two-dimensional separation of concerns (poster session). In *Addendum to the 2000 proceedings of the conference on Object-oriented programming, systems, languages, and applications (Addendum)*, pages 63–64. ACM Press, 2000.

[220] Constantinos A. Constantinides and Tzilla Elrad. Composing concerns with a framework approach. In Choukair [165], pages 133–140.

[221] Constantinos A. Constantinides, Tzilla Elrad, and Mohamed Fayad. Extending the object model to provide explicit support for crosscutting concerns. *Software Practice and Experience*, 32(7):703–734, May 2002.

[222] Constantinos A. Constantinides, Therapon Skotiniotis, and Tzilla Elrad. Providing dynamic adaptability in an aspect-oriented framework. In ECOOP-AOP01 [275].

[223] K. Cooper, L. Dai, Y. Deng, and J. Dong. Towards an aspect-oriented architectural framework. In Araújo et al. [54].

[224] Kendra Cooper, Lirong Dai, and Yi Deng. Modeling performance as an aspect: a UML based approach. In Aldawud et al. [31].

[225] P. Costanza. Dynamically scoped functions for runtime modification. In Filman et al. [296].

[226] Pascal Costanza. Separation of object identity concerns. In ECOOP-AOP00 [274].

[227] Pascal Costanza. Vanishing aspects. In OOPSLA-AOP00 [628].

[228] Pascal Costanza, Günter Kniesel, and Michael Austermann. Independent extensibility for aspect-oriented systems. In ECOOP-AOP01 [275].

[229] Pascal Costanza, Günter Kniesel, Katharina Mehner, Elke Pulvermüller, and Andreas Speck, editors. *Second Workshop on Aspect-Oriented Software Development of the German Information Society*. Institut für Informatik III, Universität Bonn, February 2002. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[230] Krzystof Czarnecki and Ulrich W. Eisenecker. *Generative Programming: Methods, Tools, and Applications*. Addison-Wesley, Boston, 2000.

[231] Krzystof Czarnecki and Ulrich W. Eisenecker. Separating the configuration aspect to support architecture evolution. In ECOOP-AOP00 [274].

[232] Krzysztof Czarnecki, Ulrich W. Eisenecker, and Patrick Steyaert. Beyond objects: Generative programming. In ECOOP-AOP97 [276].

[233] Christian Dalager, Simon Jorsal, and Eske Sort. Aspect oriented programming in JBoss 4. Master's thesis, IT University of Copenhagen, February 2004.

[234] Dangeti, Thirunavukkarasu, and Jeyabal. Runtime weaving of aspects using dynamic code instrumentation technique for building adaptive software systems. In AOSD-PAT02 [50].

[235] Ayla Dantas and Paulo Borba. Adaptability aspects: An architectural pattern for structuring adaptive applications with aspects. In *Third Latin American Conference on Pattern Languages of Programming, SugarLoafPLoP'2003*, aug 2003.

[236] Ayla Dantas and Paulo Borba. Developing adaptive j2me applications using aspectj. In *Proceedins of VII Brazilian Symposium on Programming Languages, SBLP 2003*, pages 226–242, may 2003.

[237] Ayla Dantas, Paulo Borba, and Vander Alves. Using aspects to structure small devices applications. In *First Workshop on Reuse in Constrained Environments (RICE'03), OOPSLA2003*, oct 2003.

[238] Pierre-Charles David, Thomas Ledoux, and Noury M. N. Bouraqadi-Saâdani. Two-step weaving with reflection using AspectJ. In OOPSLA-AOP01 [629].

[239] Brian de Alwis, Stephan Gudmundson, Greg Smolyn, and Gregor Kiczales. Coding issues in AspectJ. In OOPSLA-AOP00 [628].

[240] Silvia de Castro Bertagnolli and Maria Lúcia Blanck Lisbôa. The frida model. In Hannemann et al. [377].

[241] A. L. de Moura, C. Ururahy, R. Cerque, and N. Rodriguez. Dynamic support for distributed auto-adaptive applications. In Akşit and Choukair [23].

[242] K. De Volder, J. Brichau, K. Mens, and T. D'Hondt. Logic meta-programming, a framework for domain-specific aspect programming languages. http://www.cs.ubc.ca/~kdvolder/binaries/cacm-aop-paper.pdf.

[243] Kris De Volder. Aspect-oriented logic meta programming. In ECOOP-AOP98 [277].

[244] Kris De Volder. Inheritance with destructive mixins for better separation of concerns. In OOPSLA-AOP00 [628].

[245] Kris De Volder. Code reuse, an essential concern in the design of aspect languages? In ECOOP-AOP01 [275].

[246] Kris De Volder and Theo D'Hondt. Aspect-oriented logic meta programming. In P. Cointe, editor, *Meta-Level Architectures and Reflection, 2nd Int'l Conf. Reflection*, volume 1616 of *LNCS*, pages 250–272. Springer Verlag, 1999.

[247] Kris De Volder, Tom Tourwé, and Johan Brichau. Logic meta programming as a tool for separation of concerns. In ECOOP-AOP00 [274].

[248] B. De Win, B. Vanhaute, and B. De Decker. Security through aspect-oriented programming. In B. De Decker, F. Piessens, J. Smits, and F. Van Herreweghen, editors, *Advances in Network and Distributed Systems Security*, volume 206 of *IFIP Conf. Proc.*, pages 125–138. Kluwer Academic Publishers, 2001.

[249] Bart De Win, Bart Vanhaute, and Bart De Decker. Towards an open weaving process. In OOPSLA-AOP01 [629].

[250] Douglas R. Dechow. Exploiting the possibilities of "weave-time" aspects in the creation of component-based ecological models. In AOSD-PAT02 [50].

[251] John Dempsey and Vinny Cahill. Aspects of system support for distributed computing. In ECOOP-AOP97 [276].

[252] Giovanni Denaro and Mattia Monga. An experience on verification of aspect properties. In *Proceedings of the 4th international workshop on Principles of software evolution*, pages 186–189. ACM Press, 2002.

[253] Morgan Deters and Ron K. Cytron. Introduction of program instrumentation using aspects. In OOPSLA-AOP01 [629].

[254] Morgan Deters, Nick Leidenfrost, and Ron K. Cytron. Translation of Java to real-time Java using aspects. In Rashid [706].

[255] Benet Devereux. Compositional reasoning about aspects using alternating-time logic. In Leavens and Clifton [514].

[256] Maja D'Hondt and Theo D'Hondt. Is domain knowledge an aspect? In ECOOP-AOP99 [278].

[257] Maja D'Hondt and Viviane Jonckers. Hybrid aspects for weaving object-oriented functionality and rule-based knowledge. In Lieberherr [520], pages 132–140.

[258] Klissiomara Dias and Paulo Borba. Padrões de projeto para estruturação de aplicações distribudas enterprise javabeans. In *Second Latin American Conference on Pattern Languages of Programming, SugarLoafPLoP'2002*, aug 2002. Published in ICMC - University of So Paulo Magazine, December, 2002, pages 55-86.

[259] Jorge L. Diaz-Herrera, Jasmin Chadha, and Neil Pittsley. Aspect-oriented UML modeling for developing embedded systems product lines. In AOSD-UML02 [52].

[260] Edsger W. Dijkstra. *A discipline of programming*. Prentice-Hall, Englewood Cliffs, New Jersey, 1976.

[261] A. Dingwal-Smith and A. Finkelstein. From requirements to monitors by way of aspects. In AOSD-EA02 [48].

[262] Lutz Dominick. Aspect of lifecycle control in a C++ framework. In ECOOP-AOP99 [278].

[263] Lutz Dominick. Instrumentation aspects require symmetric join points. In ECOOP-AOP00 [274].

[264] Lutz Dominick and Klaus Ostermann. Supporting extension of components with new paradigms. In OOPSLA-AOP00 [628].

[265] Remi Douence, Pascal Fradet, and Mario Südholt. Composition, reuse and interaction analysis of dynamic aspects. In Lieberherr [520], pages 141–150.

[266] Rémi Douence and Narendra Jussien. Non-intrusive constraint solver enhancements. In AOSD-PAT02 [50].

[267] Rémi Douence, Olivier Motelet, and Mario Südholt. A formal definition of crosscuts. In Yonezawa and Matsuoka [885], pages 170–186.

[268] Rémi Douence, Olivier Motelet, and Mario Südholt. Sophisticated crosscuts for e-commerce. In ECOOP-AOP01 [275].

[269] Jim Dowling and Vinny Cahill. The k-component architecture meta-model for self-adaptive software. In Yonezawa and Matsuoka [885], pages 81–88.

[270] Desmond D'Souza, Aamod Sane, and Alan Birchenough. First-class extensibility for UML—Packaging of profiles, stereotypes, patterns. In OOPSLA-AOP99 [631].

[271] Frederic Duclos, Jacky Estublier, and Philippe Morat. Describing and using non functional aspects in component based applications. In Kiczales [473], pages 65–75.

[272] Dominic Duggan. A mixin-based, semantics-based approach to reusing domain-specific programming languages. In Bertino [105], pages 179–200.

[273] Gary Duzan, Joseph Loyall, and Richard Schantz. Building adaptive distributed applications with middleware and aspects. In Lieberherr [520], pages 66–73.

[274] *Workshop on Aspects and Dimensions of Concerns (ECOOP 2000)*, June 2000.

[275] *Workshop on Advanced Separation of Concerns (ECOOP 2001)*, June 2001.

[276] *Workshop on Aspect Oriented Programming (ECOOP 1997)*, June 1997.

[277] *Workshop on Aspect Oriented Programming (ECOOP 1998)*, June 1998.

[278] *Int'l Workshop on Aspect-Oriented Programming (ECOOP 1999)*, June 1999.

[279] Michael Eichberg. Mda and programming languages. In *Workshop on Generative Techniques in the context of Model Driven Architecture (OOPSLA '02)*, 2002.

[280] Eric Eide, Alastair Reid, Matthew Flatt, and Jay Lepreau. Aspect weaving as component knitting: Separating concerns with knit. In ICSE-AOP01 [428].

[281] Tzilla Elrad, Mehmet Akşit, Gregor Kiczales, Karl Lieberherr, and Harold Ossher. Discussing aspects of AOP. *Comm. ACM*, 44(10):33–38, October 2001.

[282] Tzilla Elrad, Omar Aldawud, and Atef Bader. Aspect-oriented modeling: Bridging the gap between implementation and design. In *ACM SIGPLAN/SIGSOFT Conference on Generative Programming and Component Engineering (GPCE'02)*, October 2002.

[283] Tzilla Elrad, Robert E. Filman, and Atef Bader. Aspect-oriented programming. *Comm. ACM*, 44(10):29–32, October 2001.

[284] Erik Ernst. Separation of concerns and then what. In ECOOP-AOP00 [274].

[285] Erik Ernst. Syntax based modularization: Invasive or not? In OOPSLA-AOP00 [628].

[286] Erik Ernst. Loosely coupled class families. In ECOOP-AOP01 [275].

[287] Erik Ernst. Separation of concerns. In Bergmans et al. [98].

[288] Erik Ernst and David H. Lorenz. Aspects and polymorphism in AspectJ. In Akşit [22], pages 150–157.

[289] Ran Ettinger and Mathieu Verbaere. Untangling: A slice extraction refactoring. In Lieberherr [520], pages 93–101.

[290] J. Fabry. Replication as an aspect. In ECOOP-AOP98 [277].

[291] Johan Fabry, Johan Brichau, and Tom Mens. Moving code. In ECOOP-AOP01 [275].

[292] M. E. Fayad, R. S. Pradeep, and F. Seddiqui. Aspects in communications: Performance. In Aldawud et al. [31].

[293] M.E. Fayad and Anita Ranganath. Modeling aspects using software stability and UML. In Aldawud et al. [31].

[294] R. E. Filman, S. Barrett, D. D. Lee, and T. Linden. Inserting ilities by controlling communications. *Comm. ACM*, 45(1):116–122, January 2002.

[295] R. E. Filman and D. P. Friedman. Aspect-oriented programming is quantification and obliviousness. In OOPSLA-AOP00 [628].

[296] Robert Filman, Michael Haupt, and Katharina Mehner, editors. *1st Workshop on Advancing the State-of-the-Art in Run-Time Inspection (ECOOP 2003)*, July 2003.

[297] Robert Filman, Michael Haupt, Katharina Mehner, and Mira Mezini, editors. *DAW: Dynamic Aspects Workshop*, March 2004.

[298] Robert E. Filman. Retrofitting objects. In *ACM Conf. Object Oriented Programming Systems, Languages, and Applications (OOPSLA-87)*, pages 342–353, October 1987.

[299] Robert E. Filman. Achieving ilities. In Linden and Thompson [533].

[300] Robert E. Filman. Injecting ilities. In ICSE-AOP98 [429].

[301] Robert E. Filman. Managing ilities. In *Component-Based Software Engineering Workshop (ICSE)*, pages 81–85, April 1998.

[302] Robert E. Filman. Applying aspect-oriented programming to intelligent synthesis. In ECOOP-AOP00 [274].

[303] Robert E. Filman. A software architecture for intelligent synthesis environments. In *Proc. 2001 IEEE Aerospace Conference*, pages 2879–2888, March 2001.

[304] Robert E. Filman. What is aspect-oriented programming, revisited. In ECOOP-AOP01 [275].

[305] Robert E. Filman. A bibliography of aspect-oriented programming, version 1.23. Technical Report 03.01, Research Institute for Advanced Computer Science, NASA Ames Research Center, Moffett Field, California, June 2003.

[306] Robert E. Filman. Understanding AOP through the study of interpreters. In Leavens and Clifton [514].

[307] Robert E. Filman and Klaus Havelund. Realising aspects by transforming for events. In Kris De Volder, Kim Mens, Tom Mens, and Roel Wuyts, editors, *Proc. Workshop on Declarative Meta Programming to Support Software Development*, September 2002.

[308] Robert E. Filman and Klaus Havelund. Source-code instrumentation and quantification of events. In AOSD-FOAL02 [49], pages 45–49.

[309] Robert E. Filman and Klaus Havelund. The effect of AOP on software engineering, with particular attention to OIF and event quantification. In Bergmans et al. [98].

[310] Robert E. Filman, David J. Korsmeyer, and Diana D. Lee. A CORBA extension for intelligent software environments. *Advances in Engineering Software*, 31(8-9):727–732, 2000.

[311] Robert E. Filman and Diana D. Lee. Redirecting by injector. In Choukair [165], pages 141–146.

[312] Daniela Florescu, Andreas Grünhagen, and Donald Kossman. XL: An XML programming language for web service specification and composition. In *Proc. The Eleventh Int'l World Wide Web Conference*, pages 65–76, May 2002.

[313] Marcus Fontura. Dimension templates: Multi-dimensional separation of concerns in UML. In OOPSLA-AOP99 [631].

[314] Ira R. Forman. Superimposition: A form of separation of concerns for distributed systems. In OOPSLA-AOP00 [628].

[315] P. Fradet and M. Südholt. AOP: Towards a generic framework using program transformation and analysis. In ECOOP-AOP98 [277].

[316] P. Fradet and M. Südholt. An aspect language for robust programming. In ECOOP-AOP99 [278].

[317] Robert France, Geri Georg, and Indrakshi Ray. Supporting multidimensional separation of concerns. In Aldawud et al. [30].

[318] Paul Freeman. A position on an aspect oriented approach to the observer pattern. In Bergmans et al. [98].

[319] Stephen T. Frezza. The ubiquitous overlapping, dynamic concerns project vs. product. In ICSE-AOP01 [428].

[320] A. Furfaro, L. Nigro, and F. Pupo. Aspect oriented programming using actors. In Akşit and Choukair [23].

[321] Critina Gacek and Michalis Anastasopoules. Implementing product line variabilities. In *Proceedings of 2001 Symposium on Software Reusability : Putting Software Reuse in Context*, pages 109–117. ACM Press, 2001.

[322] Patrice Gahide, Noury Bouraqadi, and Laurence Duchien. Promoting component reuse by integrating aspects and contracts in an architecture model. In AOSD-PAT02 [50].

[323] Andreas Gal, Michael Franz, and Danilo Beuche. Learning from components: Fitting AOP for system software. In Coady et al. [200].

[324] Andreas Gal, Wolfgang Schroeder-Preikschat, and Olaf Spinczyk. AspectC++: Language proposal and prototype implementation. In OOPSLA-AOP01 [629].

[325] Andreas Gal and Olaf Spinczyk. Build management for AspectC++. In OOPSLA-AOP02 [630].

[326] Alessandro Garcia, Christina Chavez, Otavio Silva, Viviane Silva, and Carlos Lucena. Promoting advanced separation of concerns in intra-agent and inter-agent software engineering. In OOPSLA-AOP01 [629].

[327] Alessandro F. Garcia and Carlos J. P. de Lucena. An aspect-based object-oriented model for multi-agent systems. In ICSE-AOP01 [428].

[328] Geri Georg, Robert France, and Indrakshi Ray. Composing aspect models. In Aldawud et al. [31].

[329] Naghmeh Ghafari, Alexander Lau, Barry Pekilis, James Thai, and Rudolph Seviora. H&V consistency checking for software health monitoring. In AOSD-PAT02 [50].

[330] Holger Giese. Towards ruling component-based distributed systems with role-based modeling and cross-cutting aspects. In ICSE-AOP01 [428].

[331] Holger Giese and Alexander Vilbig. Towards aspect-oriented design and architecture. In OOPSLA-AOP00 [628].

[332] M. Glandrup. Extending C++ using the concepts of composition filters. Master's thesis, University of Twente, 1995.

[333] Maurice Glandrup and Arend Rensink. Formal foundations for reasoning about evolution. In OOPSLA-AOP01 [629].

[334] Steffen Goebel, Christoph Pohl, Simone Roettger, and Steffen Zschaler. The comquad component modelenabling dynamic selection of implementations by weaving non-functional aspects. In Lieberherr [520], pages 74–82.

[335] Joseph D. Gradecki and Nicholas Lesiecki. *Mastering AspectJ: Aspect-Oriented Programming in Java*. John Wiley and Sons, 2003.

[336] Kasper B. Graversen and Johannes Beye. Conceptual programming using roles. In AOSD-AOD02 [47].

[337] Kasper B. Graversen and Kasper Østerbye. Aspect modelling as role modelling. In OOPSLA-AOP02 [630].

[338] Kasper B. Graversen and Kasper Østerbye. Implementation of a role language for object-specific dynamic separation of concerns. In Bergmans et al. [98].

[339] Jeff Gray. Using software component generators to construct a meta-weaver framework.

In *Proceedings of the 23rd International Conference on Software Engineering*, pages 789–790. IEEE Computer Society, 2001.

[340] Jeff Gray. Handling crosscutting constraints in domain-specific modeling. In AOSD-RICX02 [51].

[341] Jeff Gray, Ted Bapty, and Sandeep Neema. Aspectifying constraints in model-integrated computing. In OOPSLA-AOP00 [628].

[342] Jeff Gray, Ted Bapty, Sandeep Neema, and James Tuck. Handling crosscutting constraints in domain-specific modeling. *Comm. ACM*, 44(10):87–93, October 2001.

[343] Jeff Gray and Suman Roychoudhury. Technique for constructing weavers from a program transformation engine. In Lieberherr [520], pages 36–45.

[344] Danny Greefhorst. Separating concerns in software logistics. In OOPSLA-AOP00 [628].

[345] Robin Green and Awais Rashid. An aspect-oriented framework for schema evolution in object-oriented databases. In AOSD-PAT02 [50].

[346] Michael Grier. Motivation for enabling separation of concerns in software product lines. In OOPSLA-AOP99 [631].

[347] Bill Griswold, Yoshikiyo Kato, and Jimmy Yuan. Aspect browser: Tool support for managing dispersed aspects. In ICSE-AOP00 [427].

[348] William G. Griswold. Coping with crosscutting software changes using information transparency. In Yonezawa and Matsuoka [885], pages 250–265.

[349] William G. Griswold, Yoshikiyo Kato, and Jimmy J. Yuan. Aspect browser: Tool support for managing dispersed aspects. In OOPSLA-AOP99 [631].

[350] William G. Griswold, Jimmy J. Yuan, and Yoshikiyo Kato. Exploiting the map metaphor in a tool for software evolution. In *Proceedings of the 23rd international conference on Software engineering*, pages 265–274. IEEE Computer Society, 2001.

[351] Iris Groher and Stefan Schulze. Generating aspect code from UML models. In Aldawud et al. [31].

[352] Iris Groher and Stefan Schulze. Generating aspect code from UML models. In Aldawud et al. [30].

[353] Paul Grunbacher, Alexander Egyed, and Nenad Medvidovic. Dimensions of concerns in requirements negotiation and architecture modeling. In ICSE-AOP00 [427].

[354] John Grundy. Aspect-oriented requirements engineering for component-based software systems. In *4th IEEE International Symposium on Requirements Engineering*, pages 84–91. IEEE Computer Society, 1999.

[355] John Grundy and Guoling Ding. Automatic validation of deployed J2EE components using aspects. In *The 17th IEEE Int'l Conf. Automated Software Engineering*, pages 47–56, September 2002.

[356] Stephan Gudmundson and Gregor Kiczales. Addressing practical software development issues in AspectJ with a pointcut interface. In ECOOP-AOP01 [275].

[357] Stephan Gudmundson and Gregor Kiczales. Data abstraction in AspectJ. In Yonezawa and Matsuoka [885], pages 270–271.

[358] Rachid Guerraoui. Strategic research directions in object-oriented programming. *ACM Computing Surveys*, 28(4):691–700, December 1996.

[359] K. Gybels. Using a logic language to express cross-cutting through dynamic joinpoints. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[360] Kris Gybels and Johan Brichau. Arranging language features for pattern-based crosscuts. In Akşit [22], pages 60–69.

[361] Naji Habra. Separation of concerns in software engineering education. In ICSE-AOP01 [428].

[362] Ouafa Hachani and Daniel Bardou. On aspect-oriented technology and object-oriented design patterns. In Hannemann et al. [377].

[363] Szaboles Hajdara. An example of generating the synchronization code of a system composed by many similar objects. In Martn et al. [556].

[364] Charles Haley, Robin Laney, and Bashar Nuseibeh. Deriving security requirements from crosscutting threat descriptions. In Lieberherr [520], pages 112–121.

[365] Imed Hammouda, Mika Pussinen, Mika Katara, and Tommi Mikkonen. UML-based approach for documenting and specializing frameworks using patterns and concern architectures. In Aldawud et al. [31].

[366] S. Hanenberg. A proposal for classifying tangled code. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[367] S. Hanenberg, R. Hirschfeld, R. Unland, and K. Kawamura. Aspect weaving: Using the base language's introspective facilities to determine join points. In Filman et al. [296].

[368] Stefan Hanenberg, Boris Bachmendo, and Rainer Unland. An object model for general-purpose aspect languages. In Bosch [125], pages 80–91.

[369] Stefan Hanenberg and Pascal Costanza. Connecting aspects in AspectJ: Strategies vs. patterns. In AOSD-PAT02 [50].

[370] Stefan Hanenberg, Robert Hirschfeld, and Rainer Unland. Morphing aspects: Incomplete woven aspects and continuous weaving. In Lieberherr [520], pages 46–55.

[371] Stefan Hanenberg and Arno Schmidmeier. Idioms for building software frameworks in AspectJ. In Coady et al. [200].

[372] Stefan Hanenberg and Rainer Unland. Grouping objects using aspect-oriented adapters. In Rashid [706].

[373] Stefan Hanenberg and Rainer Unland. Using and reusing aspects in AspectJ. In OOPSLA-AOP01 [629].

[374] Stefan Hanenberg and Rainer Unland. Roles and aspects: Similarities, differences, and synergetic potential. In *8th International Conference on Object-Oriented Information Systems*, September 2002.

[375] Stefan Hanenberg and Rainer Unland. Specifying aspect-oriented design constraints in AspectJ. In OOPSLA-AOP02 [630].

[376] Stefan Hanenberg and Rainer Unland. Parametric introductions. In Akşit [22], pages 80–89.

[377] Jan Hannemann, Ruzanna Chitchyan, and Awais Rashid, editors. *Analysis of Aspect-Oriented Software (ECOOP 2003)*, July 2003.

[378] Jan Hannemann and Gregor Kiczales. Overcoming the prevalent decomposition in legacy code. In ICSE-AOP01 [428].

[379] Jan Hannemann and Gregor Kiczales. Design pattern implementation in Java and AspectJ. In *Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications*, pages 161–173. ACM Press, 2002.

[380] Bruno Harbulot and John Gurd. Separating concerns in parallel scientific java code using aspectj. In Lieberherr [520], pages 122–131.

[381] W. H. Harrison, H. Kilov, H. L. Ossher, and I. Simmonds. From dynamic supertypes to subjects: A natural way to specify and develop systems. *IBM Systems Journal*, 35(2):244–256, June 1996.

[382] William Harrison. Composition and multiple-inheritance in OO design (Where in the madness is the method?). In OOPSLA-AOP01 [629].

[383] William Harrison, Vincent Kruskal, Harold Ossher, Peri Tarr, and Frank Tip. Common low-level support for composition and weaving. In OOPSLA-AOP02 [630].

[384] William Harrison and Harold Ossher. Subject-oriented programming—a critique of pure objects. In *Proc. 1993 Conf. Object-Oriented Programming Systems, Languages, and Applications*, pages 411–428, September 1993.

[385] William Harrison and Harold Ossher. Member-group relationships among objects. In AOSD-FOAL02 [49], pages 9–16.

[386] William Harrison, Harold Ossher, and Peri Tarr. The beginnings of a graphical environment for subject-oriented programming. In ECOOP-AOP97 [276].

[387] William Harrison, Harold Ossher, and Peri Tarr. Using delegation for software and subject composition. Technical Report RC 20946, IBM Thomas J. Watson Research Center, August 1997.

[388] William Harrison, Harold Ossher, and Peri Tarr. Asymmetrically vs. symmetrically organized paradigms for software composition. In Bergmans et al. [98].

[389] William Harrison, Peri Tarr, and Harold Ossher. A position on considerations in UML design of aspects. In AOSD-UML02 [52].

[390] Youssef Hassoun and Constantinos Constantinides. Visibility considerations and code reusability in AspectJ. In Bachmendo et al. [62].

[391] F. J. Hauck, U. Becker, M. Geier, E. Meier, U. Rastofer, and M. Steckermeier. AspectIX: A middleware for aspect-oriented programming. In ECOOP-AOP98 [277].

[392] Juan Hernandez, Michael Papathomas, Huan M. Murillo, and Fernando Sánchez. Coordinating concurrent objects: How to deal with the coordination aspect? In ECOOP-AOP97 [276].

[393] José Luis Herrero, Fernando Sánchez, Fabiola Lucio, and Miguel Toro. Introducing separation of aspects at design time. In ECOOP-AOP00 [274].

[394] José Luis Herrero, Fernando Sánchez, and Miguel Toro. Fault tolerance AOP approach. In Rashid [706].

[395] S. Herrmann and M. Mezini. Connectors for bridging mismatches between the components of a software engineering environment. *IEE Proceedings-Software*, 148(3):104–111, June 2001.

[396] S. Herrmann, M. Mezini, and K. Ostermann. Joint efforts to dispel an approaching modularity crisis. In *Sixth International Workshop on Component-Oriented Programming (WCOP)*, 2001.

[397] Stephan Herrmann. Dynamic view connectors for separating concerns in software engineering environments. In ICSE-AOP00 [427].

[398] Stephan Herrmann. Composable designs with UFA. In AOSD-UML02 [52].

[399] Stephan Herrmann. Object teams: Improving modularity for crosscutting collaborations. In Akşit and Mezini [24].

[400] Stephan Herrmann. Object confinement in object teams—reconciling encapsulation and flexible integration. In Bachmendo et al. [62].

[401] Stephan Herrmann and Mira Mezini. On the need for a unified MDSOC model: Experiences from constructing a modular software engineering environment. In OOPSLA-AOP00 [628].

[402] Stephan Herrmann and Mira Mezini. PIROL: A case study for multidimensional separation of concerns in software engineering environments. In *OOPSLA*, pages 188–207, 2000.

[403] Stephan Herrmann and Mira Mezini. Combining composition styles in the evolvable language LAC. In ICSE-AOP01 [428].

[404] Stephan Herrmann and Jan Wloka. Seven steps from an idea to industrial adoption. In Bodkin et al. [120].

[405] Dirk Heuzeroth, Welf Löwe, Andreas Ludwig, and Uwe Assmann. Aspect-oriented configuration and adaptation of component communication. In Bosch [125], pages 58–69.

[406] Rich Hilliard. Aspects, concerns, subjects, views, ... In OOPSLA-AOP99 [631].

[407] Erik Hilsdale and Jim Hugunin. Advice weaving in aspectj. In Lieberherr [520], pages 26–35.

[408] R. Hirschfeld. Advice activation in aspects. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[409] Robert Hirschfeld. Aspects - AOP with squeak. In OOPSLA-AOP01 [629].

[410] Robert Hirschfeld. Aspect-oriented programming with aspects. In Akşit and Mezini [24].

[411] Robert Hirschfeld, Ralf Lämmel, and Matthias Wagner. Design patterns and aspects—modular designs with seamless run-time integration. In Bachmendo et al. [62].

[412] Robert Hirschfeld, Kasper Østerbye, and Matthias Wagner. System integration using AOP. In Bachmendo et al. [62].

[413] Robert Hirschfeld and Matthias Wagner. Metalevel tool support in AspectS. In OOPSLA-AOP02 [630].

[414] Robert Hirschfeld, Matthias Wagner, Wolfgang Kellerer, and Christian Prehofer. AOSD for system integration and persinalization. In Bodkin et al. [120].

[415] Wai Ming Ho, Jean-Marc Jézéquel, Francois Pennaneac'h, and Noël Plouzeau. A toolkit for weaving aspect oriented UML designs. In Kiczales [473], pages 99–105.

[416] Wai-Ming Ho, Francois Pennaneach, Jean-Marc Jézéquel, and Noël Plouzeau. Aspect-oriented design with the UML. In ICSE-AOP00 [427].

[417] Ian M. Holland. Specifying reusable components using contracts. In *European Conf. on Object-Oriented Programming*, pages 287–308, Utrecht, Netherlands, 1992. Springer Verlag Lecture Notes 615.

[418] Ian M. Holland. *The Design and Representation of Object-Oriented Components*. PhD thesis, Northeastern University, 1993.

[419] D. Holmes, J. Noble, and J. Potter. Towards reusable synchronisation for object-oriented. In ECOOP-AOP98 [277].

[420] David Holmes, James Noble, and John Potter. Aspects of synchronization. In ECOOP-AOP97 [276].

[421] Pavel Hruby. Dimensions for the separation of concerns in describing software development processes. In OOPSLA-AOP99 [631].

[422] Jie Huang. Experience using AspectJ to implementation cord. In OOPSLA-AOP00 [628].

[423] Frank Hunleth, Ron Cytron, and Christopher Gill. Building customizable middleware using aspect oriented programming. In OOPSLA-AOP01 [629].

[424] Frank Hunleth and Ron K. Cytron. Footprint and feature management using aspect-oriented programming techniques. In *Proceedings of the joint conference on Languages, compilers and tools for embedded systems*, pages 38–45. ACM Press, 2002.

[425] Walter L. Hürsch and Cristina Videira Lopes. Separation of concerns. Technical Report NU-CCS-95-03, College of Computer Science, Northeastern University, Boston, MA, February 1995.

[426] Yuuji Ichisugi and Akira Tanaka. Difference-based modules: A class-independent module mechanism. In Magnusson [552], pages 62–88.

[427] *Workshop on Multi-Dimensional Separation of Concerns in Software Engineering (ICSE 2000)*, June 2000.

[428] *Workshop on Advanced Separation of Concerns in Software Engineering (ICSE 2001)*, May 2001.

[429] *Int'l Workshop on Aspect Oriented Programming (ICSE 1998)*, April 1998.

[430] John Irwin, Jean-Marc Loingtier, John R. Gilbert, Gregor Kiczales, John Lamping, Anurag Mendhekar, and Tatiana Shpeisman. Aspect-oriented programming of sparse matrix code. In *Int'l Scientific Computing in Object-Oriented Parallel Environments (ISCOPE)*, volume 1343 of *LNCS*. Springer-Verlag, 1997.

[431] Masanori Iwamoto and Jianjun Zhao. Refactoring aspect-oriented programs. In Aldawud et al. [31].

[432] H.-Arno Jacobsen. Middleware architecture design based on aspects, the open implementation metaphor and modularity. In Rashid [706].

[433] Ivar Jacobson. Use cases and aspects—working seamlessly together. *Journal of Object Technology*, 2(4), jul 2003.

[434] Doug Janzen and Kris De Volder. Navigating and querying code without getting lost. In Akşit [22], pages 178–187.

[435] Jean-Marc Jézéquel, Noël Plouzeau, Torben Weis, and Kurt Geihs. From contracts to aspects in UML designs. In AOSD-UML02 [52].

[436] Michael B. Jones. Interposition agents: Transparently interposing user code at the system interface. In *Proceedings of the 14th ACM Symposium on Operating Systems Principles*, pages 80–93, December 1993.

[437] Bo Nørregaard Jørgensen, Eddy Truyen, Frank Matthijs, and Wouter Joosen. Customization of object request brokers by application specific policies. In *Proc. Middleware'2000*, April 2000.

[438] R. K. Joshi and N. Agrawal. AspectJ implementation of dynamically pluggable filter objects in distributed environment. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[439] Susanne Jucknath. Software visualization and aspect-oriented software development. In Martn et al. [556].

[440] Matthias Jung and Ernst W. Biersack. How layering protocol software violates separation of concerns. In ECOOP-AOP00 [274].

[441] Piotr Kaminski. Applying multi-dimensional separation of concerns to software visualization. In ICSE-AOP01 [428].

[442] Toshihiro Kamiya. Soma: a paradigm to evolve software based on separation of concerns. In *Proceedings of the Workshop on Principles of Software Evolution*, pages 124–128. ACM Press, 2002.

[443] Mohamed Kandé, Omar Aldawud, Grady Booch, and Bill Harrison, editors. *Second International Workshop on Aspect-Oriented Modeling with UML (<<UML>>2002)*, September 2002.

[444] Mohamed Kande and Valentin Crettaz. Towards patterns for concern-oriented software architecture. In Aldawud et al. [31].

[445] Mohamed Mancona Kandé, Jörg Kienzle, and Alfred Strohmeier. From AOP to UML—A bottom-up approach. In AOSD-UML02 [52].

[446] Mohamed Mancona Kandé and Alfred Strohmeier. On the role of multi-dimensional separation of concerns in software architecture. In OOPSLA-AOP00 [628].

[447] Mohamed Mancona Kandé and Alfred Strohmeier. Modeling crosscutting concerns using software connectors. In OOPSLA-AOP01 [629].

[448] Matthew Kaplan. Dynamic selection: The discriminating developers way to compose. In OOPSLA-AOP00 [628].

[449] Matthew Kaplan, Harold Ossher, William Harrison, and Vincent Kruskal. Subject-oriented design and the Watson subject compiler. In *Proc. OOPSLA'96 Workshop on Subjectivity*, October 1996.

[450] Murat Karaorman, Urs Hölzle, and John Bruno. jcontractor: A reflective java library to support design by contract. In P. Cointe, editor, *Meta-Level Architectures and Reflection, 2nd Int'l Conf. Reflection*, volume 1616 of *LNCS*, pages 175–196, Berlin, 1999. Springer Verlag.

[451] E. P. Kasten, P. K. McKinley, S. M. Sadjadi, and R. E. K. Stirewalt. Separating introspection and intercession to support metamorphic distributed systems. In Akşit and Choukair [23].

[452] Mika Katara. Superposing UML class diagram. In AOSD-UML02 [52].

[453] Mika Katara and Shmuel Katz. Architectural views of aspects. In Akşit [22], pages 1–10.

[454] Mika Katara and Tommi Mikkonen. Refinements and aspects in uml. In Kandé et al. [443].

[455] Shmuel Katz. A superimposition control construct for distributed systems. *ACM Trans. Prog. Lang. Sys.*, 15(2):337–356, April 1993.

[456] Shmuel Katz and Yossi Gil. Aspects and superimpositions. In ECOOP-AOP99 [278].

[457] P. Kellomaeki. Formal aspects for distributed systems. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[458] P. Kellomaki. Composing distributed systems from reusable aspects of behaviour. In Akşit and Choukair [23].

[459] Pertti Kellomäki. A formal basis for aspect-oriented specification with superposition. In AOSD-FOAL02 [49], pages 27–32.

[460] Pertti Kellomäki and Tommi Mikkonen. Separating product variance and domain concepts in the specification of software product lines. In ECOOP-AOP00 [274].

[461] Elizabeth A. Kendall. Agent roles and aspects. In ECOOP-AOP98 [277].

[462] Elizabeth A. Kendall. Aspect-oriented programming for role models. In ECOOP-AOP99 [278].

[463] Elizabeth A. Kendall. Role model designs and implementations with aspect-oriented programming. In *Proceedings of the 1999 ACM*

SIGPLAN conference on Object-oriented programming, systems, languages, and applications, pages 353–369. ACM Press, 1999.

[464] Liz Kendall. Reengineering for separation of concerns. In ICSE-AOP00 [427].

[465] P. Kenens, S. Michiels, F. Matthijs, B. Robben, E. Truyen, B. Vanhaute, W. Joosen, and P. Verbaeten. An AOP case with static and dynamic aspects. In ECOOP-AOP98 [277].

[466] Mik Kersten. AO tools: State of the (AspectJ) art and open problems. In OOPSLA-AOP02 [630].

[467] Mik Kersten. Tool requirements for commercial development with AspectJ. In Bodkin et al. [120].

[468] Mik A. Kersten and Gail C. Murphy. Atlas: A case study in building a web-based learning environment using aspect-oriented programming. In *Proc. ACM Conf. Object-oriented Programming, Systems, Languages, and Applications*, pages 340–352. ACM, 1999.

[469] Emre Kiciman and Armando Fox. Separation of concerns in networked service composition. In ICSE-AOP01 [428].

[470] G. Kiczales. Aspect-oriented programming. *ACM Computing Surveys*, 28(4es):154, 1996.

[471] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold. Getting started with AspectJ. *Comm. ACM*, 44(10):59–65, October 2001.

[472] G. Kiczales, E. Hilsdale, J. Hugunin, M. Kersten, J. Palm, and W. G. Griswold. An overview of AspectJ. In J. L. Knudsen, editor, *Proc. ECOOP 2001, LNCS 2072*, pages 327–353, Berlin, June 2001. Springer-Verlag.

[473] Gregor Kiczales, editor. *Proc. 1st Int' Conf. on Aspect-Oriented Software Development (AOSD-2002)*. ACM Press, April 2002.

[474] Gregor Kiczales. The key to aop. *Software Development*, December 2003.

[475] Gregor Kiczales. Testing the waters. *Software Development*, November 2003.

[476] Gregor Kiczales and Erik Hilsdale. Aspect-oriented programming. In *Proceedings of the 8th European Software Engineering Conference held jointly with 9th Acm Sigsoft Symposium on Foundations of Software Engineering*, page 313. ACM Press, 2001.

[477] Gregor Kiczales, Jim Hugunin, Mik Kersten, John Lamping, Cristina Lopes, and William G. Griswold. Semantics-based crosscutting in AspectJ. In ICSE-AOP00 [427].

[478] Gregor Kiczales, John Lamping, Anurag Mendhekar, Chris Maeda, Cristina Lopes, Jean-Marc Loingtier, and John Irwin. Aspect-oriented programming. In Mehmet Akşit and Satoshi Matsuoka, editors, *11th Europeen Conf. Object-Oriented Programming*, volume 1241 of *LNCS*, pages 220–242. Springer Verlag, 1997.

[479] Jörg Kienzle and Rachid Guerraoui. Aspect oriented software development AOP: Does it make sense? The case of concurrency and failures. In Magnusson [552], pages 37–61.

[480] Jörg Kienzle, Yang Yu, and Jie Xiong. On composition and reuse of aspects. In Bergmans et al. [98].

[481] Jörg Kienzle, Yang Yu, and Jie Xiong. On composition and reuse of aspects. In Leavens and Clifton [514].

[482] Howard Kim and Siobhán Clarke. The relevance of AOP to an applications programmer in an EJB environment. In AOSD-PAT02 [50].

[483] Doug Kimmelman. Multidimensional tree-structured spaces for separation of concerns in software development environments. In OOPSLA-AOP99 [631].

[484] Tomoij Kishi and Natsuko Noda. Aspect-oriented analysis for product line architecture. In OOPSLA-AOP00 [628].

[485] Tomoji Kishi and Natsuko Noda. Analyzing concerns used in analysis/design techniques. In ICSE-AOP01 [428].

[486] Günter Kniesel, Pascal Costanza, and Michael Austermann. JMangler—a framework for load-time transformation of Java class files. In *First IEEE Int'l Workshop on Source Code Analysis and Manipulation (SCAM 2001)*, November 2001.

[487] Jorgen Knudsen. Aspect-oriented programming in BETA using the fragment system. In ECOOP-AOP99 [278].

[488] Sergei Kojarski, Karl Lieberherr, David H. Lorenz, and Robert Hirschfeld. Aspectual reflection. In Bergmans et al. [98].

[489] P. Koopmans. On the design and realization of the sina compiler. Master's thesis, University of Twente, 1995.

[490] P. Koopmans. Sina user's guide and reference manual. Technical report, Dept. of Computer Science, University of Twente, 1995.

[491] Jan Kort and Ralf Lämmel. Separating representation concerns in re-engineering applications of rewriting. In Bachmendo et al. [62].

[492] Vincent Kruskal. A blast from the past: Using P-EDIT for multidimensional editing. In ICSE-AOP00 [427].

[493] Uirá Kulesza and Dilma M. Silva. Reengineering of the JAWS web server design using aspect-oriented programming. In ECOOP-AOP00 [274].

[494] Mario Kupries and Erika Horn. An architecture-based approach substantiating the aspect of interagent connection in platforms. In Choukair [165], pages 127–132.

[495] Ivan Kurtev. Transformation of XML concerns into applications using AOP-techniques. In AOSD-RICX02 [51].

[496] Ramnivas Laddad. Aspect-oriented refactoring part 1: Overview and process. Technical report, TheServerSide.com, 2003.

[497] Ramnivas Laddad. *AspectJ in Action*. Manning, 2003.

[498] Robert Laddaga, Paul Robertson, and Howie Shrobe. Aspects of the real-world. In OOPSLA-AOP01 [629].

[499] Ralf Laemmel, Eelco Visser, and Joost Visser. Strategic programming meets adaptive programming. In Akşit [22], pages 168–177.

[500] S. Laemmermann and E. Tyugu. A specification logic for dynamic composition of services. In Choukair [165], pages 157–162.

[501] Donal Lafferty and Vinny Cahill. Real world evaluation of Aspect-Oriented Programming with Iguana. In ECOOP-AOP00 [274].

[502] Donal Lafferty and Vinny Cahill. Language-independent aspect-oriented programming. In *Proceedings of the 18th ACM conference on Object-oriented programming, systems, languages, and applications*, 2003.

[503] Albert Lai and Gail C. Murphy. The structure of features in Java code: An exploratory investigation. In OOPSLA-AOP99 [631].

[504] Albert Lai and Gail C. Murphy. Capturing concerns with conceptual modules. In ICSE-AOP01 [428].

[505] Albert Lai, Gail C. Murphy, and Robert J. Walker. Separating concerns with HyperJ: An experience report. In ICSE-AOP00 [427].

[506] Ralf Lämmel. Declarative aspect-oriented programming. In *ACM SIGPLAN Workshop on Partial Evaluation and Semantics-Based Program Manipulation (PEPM 99)*, 1999.

[507] Ralf Lämmel. A semantical approach to method-call interception. In Kiczales [473], pages 41–55.

[508] Ralf Lämmel. Adding superimposition to a language semantics. In Leavens and Clifton [514].

[509] Ralf Lämmel, Günter Riedewald, and Wolfgang Lohmann. Adaptation of functional object programs. In ECOOP-AOP99 [278].

[510] John Lamping. The interaction of components and aspects. In ECOOP-AOP97 [276].

[511] John Lamping. The role of base in aspect-oriented programming. In ECOOP-AOP99 [278].

[512] John Lamping. The role of the base in aspect-oriented programming. In OOPSLA-AOP99 [631].

[513] David Larochelle, Karl Scheidt, and Kevin Sullivan. Join point encapsulation. In Bergmans et al. [98].

[514] Gary T. Leavens and Curtis Clifton, editors. *FOAL: Foundations of Aspect-Oriented Languages*, March 2003.

[515] Gary T. Leavens, Curtis Clifton, and Ralf Lämmel, editors. *FOAL: Foundations Of Aspect-Oriented Languages*, March 2004.

[516] Diana Lee and Robert Filman. Verification of compositional software architectures. In Linden and Thompson [533].

[517] H. Leroux, C. Mingins, and A. Requile-Romanczuk. Jacot: A uml-based tool for the runtime-inspection of concurrent java programs. In Filman et al. [296].

[518] B. Lewis. Recording events to analyze programs. In Filman et al. [296].

[519] Harry C. Li, Shriram Krishnamurthi, and Kathi Fisler. Interfaces for modular feature verification. In *Proc. 17th IEEE Int'l Conf. on Automated Software Engineering*, pages 195–204, September 2002.

[520] Karl Lieberherr, editor. *Proc. 3rd Int' Conf. on Aspect-Oriented Software Development (AOSD-2004)*. ACM Press, March 2004.

[521] Karl Lieberherr, David Lorenz, and Mira Mezini. Programming with Aspectual Components. Technical Report NU-CCS-99-01, College of Computer Science, Northeastern University, Boston, MA, March 1999.

[522] Karl Lieberherr, David H. Lorenz, and Johan Ovlinger. Aspectual collaborations: Combining modules and aspects. *The Computer Journal*, 46(5):542–565, September 2003.

[523] Karl Lieberherr, David H. Lorenz, and Pengcheng Wu. A case for statically executable advice: Checking the law of demeter with AspectJ. In Akşit [22], pages 40–49.

[524] Karl Lieberherr, Doug Orleans, and Johan Ovlinger. Aspect-oriented programming with adaptive methods. *Comm. ACM*, 44(10):39–41, October 2001.

[525] Karl J. Lieberherr. Object-oriented programming with class dictionaries. *Journal on Lisp and Symbolic Computation*, 1(2):185–212, 1988.

[526] Karl J. Lieberherr. Component enhancement: An adaptive reusability mechanism for groups of collaborating classes. In J. van Leeuwen, editor, *Information Processing '92, 12th World Computer Congress*, pages 179–185, Madrid, Spain, 1992. Elsevier.

[527] Karl J. Lieberherr. *Adaptive Object-Oriented Software: the Demeter Method with Propagation Patterns*. PWS Publishing Company, Boston, 1996.

[528] Karl J. Lieberherr and Ian Holland. Assuring good style for object-oriented programs. *IEEE Software*, pages 38–48, September 1989.

[529] Karl J. Lieberherr, Walter Hürsch, Ignacio Silva-Lepe, and Cun Xiao. Experience with a graph-based propagation pattern programming tool. In Gene Forte and Nazim Madhavji, editors, *Int'l Workshop on CASE*, pages 114–119, Montréal, Canada, 1992. IEEE Computer Society.

[530] Karl J. Lieberherr and Doug Orleans. Preventive program maintenance in Demeter/Java (research demonstration). In *Int'l Conf. Software Engineering*, pages 604–605, Boston, MA, 1997. ACM Press.

[531] Karl J. Lieberherr, Ignacio Silva-Lepe, and Cun Xiao. Adaptive object-oriented programming using graph-based customization. *Comm. ACM*, 37(5):94–101, May 1994.

[532] Karl J. Lieberherr and Cun Xiao. Object-oriented software evolution. *IEEE Trans. Soft. Eng.*, 19(4):313–343, April 1993.

[533] Ted Linden and Craig Thompson, editors. *OMG-DARPA Workshop on Compositional Software Architectures*, January 1998.

[534] Jean Marie Lions, Didier Simoneau, Gilles Pitette, and Imed Moussa. Extending opentool/uml using metamodeling: An aspect-oriented programming case study. In Kandé et al. [443].

[535] Martin Lippert and Cristina Videira Lopes. A study on exception detecton and handling using aspect-oriented programming. In *Proceedings of the 22nd International Conference on Software Engineering*, pages 418–427. ACM Press, 2000.

[536] D. Lohmann and J. Ebert. A generalization of the hyperspace approach using meta-models. In Araújo et al. [54].

[537] Christina Lopes, Erik Hilsdale, Jim Hugunin, Mik Kersten, and Gregor Kiczales. Illustrations of crosscutting. In ECOOP-AOP00 [274].

[538] Crista Videira Lopes and Gregor Kiczales. Recent developments in AspectJ. In ECOOP-AOP98 [277].

[539] Cristina Videira Lopes. Graph-based optimizations for parameter passing in remote invocations. In Luis-Felipe Cabrera and Marvin Theimer, editors, *4th Int'l Workshop on Object Orientation in Operating Systems*, pages 179–182, Lund, Sweden, August 1995. IEEE Computer Society Press.

[540] Cristina Videira Lopes. Adaptive parameter passing. In *2nd Int'l Symposium on Object Technologies for Advanced Software*, pages 118–136, Kanazawa, Japan, March 1996. Springer-Verlag.

[541] Cristina Videira Lopes. *D: A Language Framework for Distributed Programming.* PhD thesis, College of Computer Science, Northeastern University, 1997.

[542] Cristina Videira Lopes and Gregor Kiczales. D: A language framework for distributed programming. Technical Report SPL-97-010, Palo Alto Research Center, 1997.

[543] Cristina Videira Lopes and Karl Lieberherr. AP/S++: Case-study of a MOP for purposes of software evolution. In G. Kiczales, editor, *Reflection '96*, San Francisco, April 1996.

[544] Cristina Videira Lopes and Karl J. Lieberherr. Abstracting process-to-function relations in concurrent object-oriented applications. In Tokoro and Pareschi [829], pages 81–99.

[545] Critina Lopes, Paul Dourish, David Lorenz, and Karl Lieberherr. Beyond aop: Toward naturalistic programming. In *Onward! in the 18th ACM conference on Object-oriented programming, systems, languages, and applications*, 2003.

[546] D. H. Lorenz. Visitor beans: An aspect-oriented pattern. In ECOOP-AOP98 [277].

[547] David H. Lorenz and Yvonne Coady, editors. *ACP4IS: Aspects, Components, and Patterns for Infrastructure Software*, March 2004.

[548] N. Loughran and A. Rashid. Mining aspects. In AOSD-EA02 [48].

[549] Neil Loughran and Awais Rashid. Relational database support for aspect-oriented programming. In Akşit and Mezini [24].

[550] Tina Low. Designing, modelling and implementing a toolkit for aspect-oriented tracing (TAST). In AOSD-UML02 [52].

[551] C. P. Lunau. Is composition of metaobjects = aspect oriented programming. In ECOOP-AOP98 [277].

[552] B. Magnusson, editor. *ECOOP 2002—Object-Oriented Programming: 16th European Conference, LNCS 2374.* Springer Verlag, June 2002.

[553] Tom Mahieu, Bart Vanhaute, Karel De Vlaminck, Gerda Janssens, and Wouter Joosen. Using AOP to build complex data centric component frameworks. In OOPSLA-AOP00 [628].

[554] Jacques Malenfant, Maria-Teresa Segarra, and Franoise André. Dynamic adaptability: The MoléNE experiment. In Yonezawa and Matsuoka [885], pages 110–117.

[555] Frank Manola. Providing systemic properties (ilities) and quality of service in component-based systems. Technical report, OBJS, 1999.

[556] Pedro Jos´ Clemente Martn, Sergio Lujan Mora, Miguel Ángel Pérez Toledano, Hans Resiser, and Sergio Soares, editors. *The 13th Workshop for PhD Students in Object-Oriented Systems (ECOOP 2003)*, July 2003.

[557] T. Massoni, V. Alves, S. Soares, and P. Borba. Pdc: Persistent data collections pattern. In *First Latin American Conference on Pattern Languages of Programming — SugarLoafPLoP*, October 2001.

[558] Tiago Massoni, Augusto Sampaio, and Paulo Borba. Progressive implementation of aspects. In OOPSLA-AOP01 [629].

[559] Hidehiko Masuhara and Gregor Kiczales. A modeling framework for aspect-oriented mechanisms; draft. http://www.cs.ubc.ca/

[560] Hidehiko Masuhara, Gregor Kiczales, and Chris Dutchyn. Compilation semantics of aspect-oriented programs. In AOSD-FOAL02 [49], pages 17–26.

[561] Frank Matthijs, Wouter Joosen, Bart Vanhaute, Bert Robben, and Pierre Verbaeten. Aspects should not die. In ECOOP-AOP97 [276].

[562] Sean McDirmid and Wilson C. Hsieh. Aspect-oriented programming with Jiazzi. In Akşit [22], pages 70–79.

[563] David L. McReynolds, Sheryl L. Duggins, Doreen L. Galli, and John H. Mayer. Distributed characteristics of subject oriented programming: An evaluation with the process and object paradigms. In *Proceedings of the 37th Annual Southeast Regional Conference (CD-ROM)*, page 19. ACM Press, 1999.

[564] Katharina Mehner and Awais Rashid. Towards a standard interface for runtime inspection in aop environments. In OOPSLA-AOP02 [630].

[565] Katharina Mehner and Annika Wagner. On the role of method families in aspect-oriented programming. In ECOOP-AOP99 [278].

[566] François Mekerke, Geri Georg, Robert France, and Roger Alexander. Tool support for aspect-oriented design. In J.-M. Bruel and Z. Bellahséne, editors, *Advances in Object-Oriented Information Systems OOIS 2002 Workshops, LNCS 2426*, pages 280–289, September 2002.

[567] Stephen J. Mellor. A framework for aspect-oriented modeling. In Aldawud et al. [31].

[568] Juri Memmer. Designing with Cosmos. In AOSD-AOD02 [47].

[569] Juri Memmert. Application development in Java: From OOP to SOP. In ICSE-AOP00 [427].

[570] Juri Memmert. Separation of concerns at the source. In ICSE-AOP01 [428].

[571] Juri Memmert. Is aosd really worth is? In Bodkin et al. [120].

[572] Anurag Mendhekar, Gregor Kiczales, and John Lamping. RG: A case-study for aspect-oriented programming. Technical Report SPL-97-009, Palo Alto Research Center, 1997.

[573] K. Mens. Architectural aspects. In AOSD-EA02 [48].

[574] Kim Mens. Multiple cross-cutting architectural views. In ICSE-AOP00 [427].

[575] Kim Mens, Tom Mens, and Michel Wermelinger. Supporting software evolution with intentional software views. In *Proceedings of the Workshop on Principles of Software Evolution*, pages 138–142. ACM Press, 2002.

[576] Tom Mens and Michel Wermelinger. Separation of concerns for software evolution. *Journal of Software Maintenance and Evolution: Research and Practice*, 14(5):311–315, 2002.

[577] C. Mesquita, S. Barbosa, and C. de Lucena. Towards the identification of concerns in personalization mechanisms via scenarios. In AOSD-EA02 [48].

[578] Wolfgang De Meuter. Monads as a theoretical foundation for AOP. In ECOOP-AOP97 [276].

[579] M. Mezini. Dynamic metaclass construction for an explicit specialization interface. In G. Kiczales, editor, *Reflection '96*, pages 203–219, San Francisco, April 1996.

[580] M. Mezini. Incremental redefinition of open implementations. In C. Zimmermann, editor, *Advances in Object-Oriented Metalevel Architectures and Reflection*, pages 265–290. CRC Press, Boca Raton, 1996.

[581] M. Mezini. Dynamic object modification without name collisions. In Mehmet Akşit and Satoshi Matsuoka, editors, *11th Europeen Conf. Object-Oriented Programming*, volume 1241 of *LNCS*, pages 190–219. Springer Verlag, 1997.

[582] M. Mezini and K. Ostermann. Modules for crosscutting models. In *8th International Conference on Reliable Software Technologies (Ada-Europe '03)*. Springer-Verlag Lecture Notes in Computer Science, June 2003.

[583] Mira Mezini. Maintaining the consistency of class libraries during their evolution. In *Proc. 1997 ACM SIGPLAN conference on Object-Oriented Programming Systems, Languages and Applications*, pages 1–21. ACM Press, 1997.

[584] Mira Mezini. *Variation-Oriented Programming Beyond Classes and inheritance*. PhD thesis, University of Siegen, 1997.

[585] Mira Mezini and Karl Lieberherr. Adaptive plug-and-play components for evolutionary software development. In C. Chambers,

editor, *OOPSLA*, pages 97–116. ACM, October 1998.

[586] Mira Mezini and Klaus Ostermann. Integrating independent components with on-demand remodularization. In *Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications*, pages 52–67. ACM Press, 2002.

[587] Mira Mezini, Linda Seiter, and Karl Lieberherr. Component integration with pluggable composite adapters. In Mehmet Akşit, editor, *Symposium on Software Architectures and Component Technology: The State of the Art in Research and Practice*. Kluwer Academic Publishers, October 2001.

[588] T. Mikkonen. On objects, aspects, and specifications addressing their collaboration. In AOSD-EA02 [48].

[589] H. Mili, H. Mcheick, and S. Sadou. Distribution and aspects. In Akşit and Choukair [23].

[590] H. Mili, H. Mcheick, and S. Sadou. Distribution and aspects. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[591] Hafedh Mili, Joumana Dargham, and Salah Bendelloul. Separation of concerns and typing: A first stab. In OOPSLA-AOP99 [631].

[592] Hafedh Mili, William Harrison, and Harold Ossher. Supporting subject-oriented programming in smalltalk. In *TOOLS USA 96*, August 1996.

[593] Naftaly Minsky. Law-governed regularities in object systems; part 1: Principles. *Theory and Practice of Object Systems (TOPAS)*, 2(4), 1996.

[594] Naftaly Minsky and Partha Pal. Law-governed regularities in object systems; part 2: the eiffel case. *In Theory and Practice of Object Systems*, 3(2), 1997.

[595] Mattia Monga. Concern-specific aspect-oriented programming with Malaj. In ICSE-AOP00 [427].

[596] Alexandra Weber Morales and John Reitano. The 15% solution. *Software Development*, 10(5):34–38, May 2002.

[597] Ana Moreira, João Araújo, and Isabel Brito. Crosscutting quality attributes for requirements engineering. In *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering*, pages 167–174. ACM Press, 2002.

[598] M. Mousavi, G. Russello, M. Chaudron, M. A. Reniers, T. Basten, A. Corsaro, S. Shukla, R. Gupta, and D. Schmid. Aspects + GAMMA = AspectGAMMA: A formal framework for aspect-oriented specification. In AOSD-EA02 [48].

[599] Jurgen K. Muller. Aspect-design in the building-block method. In ECOOP-AOP97 [276].

[600] Gail C. Murphy, Albert Lai, Robert J. Walker, and Martin P. Robillard. Separating features in source code: An exploratory study. In *Proc. 23rd Int'l Conf. Software Engineering*, pages 275–284. IEEE Computer Society, 2001.

[601] Gail C. Murphy, Robert J. Walker, and Elisa L. A. Baniassad. Evaluating emerging software development technologies: Lessons learned from assessing aspect-oriented programming. *IEEE Transactions on Software Engineering*, 25(4):438–455, 1999.

[602] Gail C. Murphy, Robert J. Walker, Elisa L. A. Baniassad, Martin P. Robillard, Albert Lai, and Mik A. Kersten. Does aspect-oriented programming work? *Comm. ACM*, 44(10):75–77, October 2001.

[603] István Nagy, Mehmet Aksit, and Lodewijk Bergmans. Composition graphs, a foundation for reasoning about aspect-oriented composition. In Leavens and Clifton [514].

[604] Shin Nakajima. Separation of concerns in early stage of framework development. In OOPSLA-AOP99 [631].

[605] A. Navasa, M. A. Perez, J. M. Murillo, and J. Hernandez. Aspect-oriented software architecture: A structural perspective. In AOSD-EA02 [48].

[606] A. Navasa, M. A. Pérez, and J.M. Murillo. Using an adl to design aspect oriented systems. In Martn et al. [556].

[607] Amparo Navasa, Miguel A. Pérez, and Juan M. Murillo. Developing component based systems using AOP concepts. In ECOOP-AOP01 [275].

[608] R. D. Nebbe. Coordination and composition: The two paradigms underlying AOP? In ECOOP-AOP98 [277].

[609] Andrey Nechypurenko. Using design patterns to improve aspect reusability and dynamics. In AOSD-PAT02 [50].

[610] Torsten Nelson, Paulo Alencar, and Donald Cowan. Verifying multiple-perspective composition. In OOPSLA-AOP00 [628].

[611] Torsten Nelson, Donald Cowan, and Paulo Alencar. Towards a formal model of object-oriented hyperslices. In OOPSLA-AOP99 [631].

[612] Torsten Nelson, Donald Cowan, and Paulo Alencar. Formal verification of a bounded buffer with three separate concerns. In ICSE-AOP01 [428].

[613] Torsten Nelson, Donald Cowan, and Paulo Alencar. Supporting formal verification of crosscutting concerns. In Yonezawa and Matsuoka [885], pages 153–169.

[614] Christian Nentwich, Wolfgang Emmerich, and Anthony Finkelstei. Edit, compile, debug—From hacking to distributed engineering. In AOSD-AOD02 [47].

[615] P. Netinant, C. A. Constantinides, T. Elrad, and M. E. Fayad. Aspect-oriented frameworks (poster session): the design of adaptable operating systems. In *Addendum to the 2000 proceedings of the conference on Object-oriented programming, systems, languages, and applications (Addendum)*, pages 61–62. ACM Press, 2000.

[616] Paniti Netinant, Tzilla Elrad, and Mohamed E. Fayad. A layered approach to building open aspect-oriented systems: A framework for the design of on-demand system demodularization. *Comm. ACM*, 44(10):83–85, October 2001.

[617] Elissa Newman. Localizing views for separation of concerns. In ICSE-AOP01 [428].

[618] Elissa Newman and William L. Scherlis. Toward query-based constraints. In Bergmans et al. [98].

[619] Oscar Nierstrasz and Franz Achermann. Separation of concerns through unification of concepts. In ECOOP-AOP00 [274].

[620] Oscar Nierstrasz and Franz Achermann. Supporting compositional styles for software evolution. In *Proceedings Int'l Symposium on Principles of Software Evolution (ISPSE 2000)*, pages 11–19. IEEE, Nov 1-2 2000.

[621] Muga Nishizawa, Shigeru Chiba, and Michiaki Tatsubori. Remote pointcut – a language construct for distributed aop. In Lieberherr [520], pages 7–15.

[622] Natsuko Noda and Tomoji Kishi. On aspect-oriented design—Applying "multi-dimensional separation of concerns" on designing quality attributes. In OOPSLA-AOP99 [631].

[623] Natsuko Noda and Tomoji Kishi. Implementing design patterns using advanced separation of concerns. In OOPSLA-AOP01 [629].

[624] Martin E. Nordberg III. Aspect-oriented dependency inversion. In OOPSLA-AOP01 [629].

[625] A. Noutash, M. van Sinderen, M. Akşit, and L. Bergmans. Qos provisioning in corba by introducing aspect-oriented transport reflection point. In *ECOOP'2000 Workshop on Quality of Service*, June 2000.

[626] Bashar Nuseibeh. Crosscutting requirements. In Lieberherr [520], pages 3–4.

[627] Brain Odgers and Simon Thompson. Aspect-oriented process engineering. In ECOOP-AOP99 [278].

[628] *Workshop on Advanced Separation of Concerns (OOPSLA 2000)*, October 2000.

[629] *Workshop on Advanced Separation of Concerns in Object-Oriented Systems (OOPSLA 2001)*, October 2001.

[630] *Workshop on Tools for Aspect-Oriented Software Development (OOPSLA 2002)*, November 2002.

[631] *Workshop on Multi-Dimensional Separation of Concerns (OOPSLA 1999)*, November 1999.

[632] Graham O'Regan. Introduction to aspect-oriented programming. *O'Reilly On-Java.com*, January 2004.

[633] Doug Orleans. Separating behavioral concerns with predicate dispatch, or, if statement considered harmful. In OOPSLA-AOP01 [629].

[634] Doug Orleans. Incremental programming with extensible decisions. In Kiczales [473], pages 56–64.

[635] Doug Orleans and Karl Lieberherr. DJ: Dynamic adaptive programming in Java. In Yonezawa and Matsuoka [885], pages 73–80.

[636] H. Ossher, M. Kaplan, A. Katz, W. Harrison, and V. Kruskal. Specifying subject-oriented composition. *Theory and Practice of Object Systems*, 2(3), 1996.

[637] H. Ossher and P. Tarr. Operation-level composition: A case in (join) point. In ECOOP-AOP98 [277].

[638] H. Ossher and P. Tarr. Multi-dimensional separation of concerns using hyperspaces. Technical Report 21452, IBM Research Report, April 1999.

[639] H. Ossher and P. Tarr. Multi-dimensional separation of concerns and the hyperspace approach. In *Proc. Symposium on Software Architectures and Component Technology: The State of the Art in Software Development*. Kluwer, 2000.

[640] H. Ossher and P. Tarr. The shape of things to come: Using multi-dimensional separation of concerns with Hyper/J to (re)shape evolving software. *Comm. ACM*, 44(10):43–50, October 2001.

[641] Harold Ossher and William Harrison. Combination of inheritance hierarchies. In *Proc. 1992 Conf. Object-Oriented Programming Systems, Languages, and Applications*, October 1992.

[642] Harold Ossher, William Harrison, Frank Budinsky, and Ian Simmonds. Subject-oriented programming: Supporting decentralized development of objects. In *Proc. 7th IBM Conf. Object-Oriented Technology*, July 1994.

[643] Harold Ossher, William Harrison, and Peri Tarr. Software engineering tools and environments: A roadmap. In *Proceedings of the Conference on the Future of Software Engineering*, pages 261–277. ACM Press, 2000.

[644] Harold Ossher, Matthew Kaplan, William Harrison, Alexander Katz, and Vincent Kruskal. Subject-oriented composition rules. In *Proc. 1995 Conf. Object-Oriented Programming Systems, Languages, and Applications*, October 1995.

[645] Harold Ossher and Peri Tarr. Using subject-oriented programming to overcome common problems in object-oriented software development/evolution. In *Proc. 21st Int'l Conf. Software Engineering*, pages 687–688. IEEE Computer Society Press, 1999.

[646] Harold Ossher and Peri Tarr. Hyper/J: Multi-dimensional separation of concerns for Java. In *Proc. 22nd Int'l Conf. Software Engineering*, pages 734–737. ACM Press, 2000.

[647] Harold Ossher and Peri Tarr. On the need for on-demand remodularization. In ECOOP-AOP00 [274].

[648] Harold Ossher and Peri Tarr. Some micro-reuse challenges. In ECOOP-AOP01 [275].

[649] Harold Ossher and Perri Tarr. Multi-dimensional separation of concerns in hyperspace. In ECOOP-AOP99 [278].

[650] Harold Ossher and Petri Tarr. Hyper/J: Multi-dimensional separation of concerns for Java. In *Proc. 23rd Int'l Conf. on Software Engineering*, pages 729–730. IEEE Computer Society, 2001.

[651] Klaus Ostermann. Dynamically composable collaborations with delegation layers. In Magnusson [552], pages 89–110.

[652] Klaus Ostermann and Günter Kniesel. Independent extensibility—An open challange for AspectJ and Hyper/J. In ECOOP-AOP00 [274].

[653] Klaus Ostermann and Mira Mezini. Object-oriented composition is tangled. In ECOOP-AOP01 [275].

[654] Klaus Ostermann and Mira Mezini. Object-oriented composition untangled. In *Proc.*

OOPSLA '01 Conf. Object Oriented Programming Systems Languages and Applications, pages 283–299. ACM Press, 2001.

[655] Klaus Ostermann and Mira Mezini. Conquering aspects with Caesar. In Akşit [22], pages 90–99.

[656] Johan Ovlinger. From aspect oriented model to implementation watch out for impedance and mismatch. In Aldawud et al. [30].

[657] Johan Ovlinger, Karl Lieberherr, and David Lorenz. Aspects and modules combined. Technical Report NU-CCS-02-03, College of Computer Science, Northeastern University, Boston, MA, March 2002.

[658] J. Andrés Díaz Pace and Marcelo R. Campo. Analyzing the role of aspects in software design. *Comm. ACM*, 44(10):66–73, October 2001.

[659] J. Andrés Díaz Pace, Mohamed E. Fayad, and Marcelo R. Campo. A language for simulation: Bringing separation to the front. In ECOOP-AOP00 [274].

[660] J. Andrés Díaz Pace, F. Trilnik, and Marcelo R. Campo. How to handle interacting concerns? In OOPSLA-AOP00 [628].

[661] Jeffrey D. Palm, Kenneth M. Anderson, and Karl M. Lieberherr. Investigating the relationship between violations of the law of demeter and software maintainability. In Bergmans et al. [98].

[662] Jens Palsberg. Class-graph inference for adaptive programs. *Theory and Practice of Object Systems*, 3(2):75–85, April 1997.

[663] Jens Palsberg, Boaz Patt-Shamir, and Karl Lieberherr. A new approach to compiling adaptive programs. *Science of Computer Programming*, 29(3):303–326, 1997.

[664] Jens Palsberg, Boaz Patt-Shamir, and Karl Lieberherr. A new approach to compiling adaptive programs. In Hanne Riis Nielson, editor, *European Symposium on Programming*, pages 280–295, Linkoping, Sweden, April 1996. Springer Verlag Lecture Notes in Computer Science 1058.

[665] Jens Palsberg, Cun Xiao, and Karl Lieberherr. Efficient implementation of adaptive software. *ACM Transactions on Programming Languages and Systems (TOPLAS)*, 17(2):264–292, 1995.

[666] Thomas Panas, Jesper Andersson, and Uwe Aßmann. The editing aspect of aspects. In I. Hussain, editor, *Software Engineering and Applications (SEA2002)*, Cambridge, MA, November 2002. ACTA Press.

[667] Thomas Panas, Jonas Karlsson, and Magnus Högberg. Aspect-jEdit for inline aspect support. In Bachmendo et al. [62].

[668] J. Pang and L. Blair. An adaptive run time manager for the dynamic integration and interaction resolution of features. In Akşit and Choukair [23].

[669] Flavio De Paoli. Multidimensional separation of concerns. In ICSE-AOP00 [427].

[670] Holger Papajewski, Olaf Spinczyk, and Danilo Beuche. Success factors for aosd in commercial applications. In Bodkin et al. [120].

[671] D. L. Parnas. On the criteria to be used in decomposing systems into modules. *Comm. ACM*, 15(12):1053–1058, December 1972.

[672] Renaud Pawlak. Nature and benefits of aspect-oriented programming. In ECOOP-AOP00 [274].

[673] Renaud Pawlak, Laurence Duchien, Gerard Florin, Fabrice Legond-Aubry, Lionel Seinturier, and Laurent Martelli. A UML notation for aspect-oriented software design. In AOSD-UML02 [52].

[674] Renaud Pawlak, Lionel Seinturier, Laurence Duchien, and Gérard Florin. JAC: A flexible solution for aspect-oriented programming in Java. In Yonezawa and Matsuoka [885], pages 1–24.

[675] Renaud Pawlak and Houman Younessi. On getting use cases and aspects to work together. *Journal of Object Technology*, 3(1), 2004.

[676] Luca Pazzi. Explicit aspect composition by part-whole state charts. In ECOOP-AOP99 [278].

[677] Ilka Philippow, Matthias Riebisch, and Kai Boellert. The Hyper/UML approach for feature based software design. In Aldawud et al. [31].

[678] G. Piccinelli and L. Mokrushin. Dynamic e-service composition in Dysco. In Choukair [165], pages 88–96.

[679] G. Piccinelli and M. Stearns. Managing interaction concerns in web-service systems. In Akşit and Choukair [23].

[680] Roman Pichler, Klaus Ostermann, and Mira Mezini. On aspectualizing component models. *Software Practice and Experience*, 33(10):957–974, 2003.

[681] Scott M. Pike. Binary trees: A challenge problem for separating concerns. In ICSE-AOP01 [428].

[682] M. Pinto, M. Amor, L. Fuentes, and J. M. Troya. Collaborative virtual environment development: An aspect-oriented approach. In Choukair [165], pages 97–102.

[683] M. Pinto, M. Amor, L. Fuentes, and J. M. Troya. Run-time coordination of components: Design patterns vs. component-aspect based platforms. In ECOOP-AOP01 [275].

[684] Monica Pinto, Lidia Fuentes, Mohamed Fayad, and Jose Maria Troya. Separation of coordination in a dynamic aspect oriented framework. In Kiczales [473], pages 134–140.

[685] Eduardo Kessler Piveta and Augusto Jun Devegil. Aspects in the rational unified process analysis and design workflow. In AOSD-AOD02 [47].

[686] Eduardo Kessler Piveta and Luiz Carlos Zancanella. Aurélia: Aspect oriented programming using a reflective approach. In ECOOP-AOP01 [275].

[687] A. Popovici, G. Alonso, and T. Gross. AOP support for mobile systems. In OOPSLA-AOP01 [629].

[688] Andrei Popovici, Gustavo Alonso, and Thomas Gross. Just in time aspects. In Akşit [22], pages 100–109.

[689] Andrei Popovici, Thomas Gross, and Gustavo Alonso. Dynamic weaving for aspect-oriented programming. In Kiczales [473], pages 141–147.

[690] M. Devi Prasad. Typecasting as a new join point in AspectJ. In Martn et al. [556].

[691] M. Devi Prasad and B. D. Chaudhary. AOP support for C#. In Coady et al. [200].

[692] Christian Prehofer. Feature interactions in statechart diagrams or graphical composition of components. In Kandé et al. [443].

[693] Christian Prehofer. Graphical composition of components with feature interactions. In AOSD-UML02 [52].

[694] Jane Pryor and Natalio Bastán. A reflective architecture for the support of aspect-oriented programming in Smalltalk. In ECOOP-AOP99 [278].

[695] Elke Pulvermüller, Andreas Speck, and James O. Coplien. A version model for aspect dependency management. In Bosch [125], pages 70–79.

[696] E. Putrycz and G. Bernard. Using aspect oriented programming to build a portable load balancing service. In Akşit and Choukair [23].

[697] E. Putryez and G. Bernard. Client side reconfiguration on software components for load balancing. In Choukair [165], pages 111–116.

[698] Hridesh Rajan and Kevin Sullivan. Need for instance level aspect language with rich pointcut language. In Bergmans et al. [98].

[699] Rajeev R. Raje, Ming Zhong, and Tongyu Wang. Case study: a distributed concurrent system with AspectJ. *ACM SIGAPP Applied Computing Review*, 9(2):17–23, 2001.

[700] Rafael Ramirez and Andrew E. Santosa. An aspect-oriented methodology for concurrent and distributed applications. In Bachmendo et al. [62].

[701] A. Rashid. On to aspect persistence. In *2nd International Symposium on Generative and Component-Based Software Engineering, LNCS 2177*, pages 453–463. Springer-Verlag Lecture Notes in Computer Science, October 2000.

[702] A. Rashid. Aspect-oriented and component-based software engineering. *IEE Proceedings-Software*, 148(3):87–88, June 2001.

[703] A. Rashid and E. Pulvermueller. From object-oriented to aspect-oriented databases. In *11th International Conference on Database and Expert Systems Applications — DEXA*

2000, LNCS 1873, pages 125–134. Springer-Verlag Lecture Notes in Computer Science, September 2000.

[704] Awais Rashid. A database evolution approach for object-oriented databases. In *Proc. IEEE Int'l Conf. Software Maintenance (ICSM 2001)*, pages 561–564, November 2001.

[705] Awais Rashid. A hybrid approach to separation of concerns: The story of SADES. In Yonezawa and Matsuoka [885], pages 231–249.

[706] Awais Rashid, editor. *Workshop on Aspect-Oriented Programming and Separation of Concerns (Lancaster)*, August 2001.

[707] Awais Rashid. Weaving aspects in a persistent environment. *ACM SIGPLAN Notices*, 37(2), February 2002.

[708] Awais Rashid and Lynne Blair. Aspect-oriented programming and separation of crosscutting concerns. *The Computer Journal*, 46(5):527–528, September 2003.

[709] Awais Rashid and Ruzanna Chitchyan. Persistence as an aspect. In Akşit [22], pages 120–129.

[710] Awais Rashid and Gerald Kotonya. Risk management in component-based development: A separation of concerns perspective. In ECOOP-AOP01 [275].

[711] Awais Rashid, Ana Moreira, and Joäo Araújo. Modularisation and composition of aspectual requirements. In Akşit [22], pages 11–20.

[712] Awais Rashid and Peter Sawyer. Aspect-orientation and database systems: An effective customisation approach. *IEE Proceedings - Software*, 148(5):156–164, October 2001.

[713] U. Rastofer and F. Bellosa. Component-based software engineering for distributed embedded real-time systems. *IEE Proceedings-Software*, 148(3):99–103, June 2001.

[714] Andreas Rausch, Bernhard Rumpe, and Lucien Hoogendoorn. Aspect-oriented framework modeling. In Aldawud et al. [31].

[715] Barry Redmond and Vinny Cahill. Supporting unanticipated dynamic adaptation of application behaviour. In Magnusson [552], pages 205–230.

[716] John Regehr and Alastair Reid. Lock inference for systems software. In Coady et al. [200].

[717] A. M. Reina. Separating the navigational aspect. In Akşit and Choukair [23].

[718] A. M. Reina and J. Torres. Analysing the navigational aspect. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[719] A. M. Reina, J. Torres, and M. Toro. Aspect-oriented web development vs. non aspect-oriented web development. In Hannemann et al. [377].

[720] R. Q. Reis, C. A. Lima Reis, H. Schlebbe, and D. J. Nunes. Towards an aspect-oriented approach to improve the reusability of software process models. In AOSD-EA02 [48].

[721] Søren Top, Bo Jørgensen, Claus Thybo, and Peter Thusgaard. Meta-level architectures for fault tolerant control (FTC) in embedded software systems. In ECOOP-AOP01 [275].

[722] Karen Renaud. HERCULE: Non-invasively tracking Java component-based application activity. In Bertino [105], pages 447–471.

[723] Joel Richardson and Peter Schwarz. Aspects: extending objects to support multiple, independent roles. In *Proceedings of the 1991 ACM SIGMOD International Conference on Management of Data*, pages 298–307. ACM Press, 1991.

[724] Mark Richters and Martin Gogolla. Aspect-oriented monitoring of UML and OCL constraints. In Aldawud et al. [31].

[725] Bert Robben and Patrick Steyaert. Aspects on TV. In ECOOP-AOP00 [274].

[726] Martin P. Robillard and Gail C. Murphy. Migrating a static analysis tool to AspectJ. In OOPSLA-AOP99 [631].

[727] Martin P. Robillard and Gail C. Murphy. An exploration of a lightweight means of concern separation. In ECOOP-AOP00 [274].

[728] Martin P. Robillard and Gail C. Murphy. Analyzing concerns using class member dependencies. In ICSE-AOP01 [428].

[729] Martin P. Robillard and Gail C. Murphy. Capturing concern descriptions during program navigation. In OOPSLA-AOP02 [630].

[730] Yves Roudier and Yuuji Ichisugi. Mixin composition strategies for the modular implementation of aspect weaving. In ICSE-AOP98 [429].

[731] Isabelle Rouvellou, Stanley M. Sutton Jr., and Stefan Tai. Multidimensional separation of concerns in middleware. In ICSE-AOP00 [427].

[732] Daniel Sabbah. Aspects—from promise to reality. In Lieberherr [520], pages 1–2.

[733] Motoshi Saeki and Haruhiko Kaiya. Transformation based approach for weaving use case models in aspect-oriented requirements analysis. In Aldawud et al. [31].

[734] Kouhei Sakurai, Hidehiko Masuhara, Naoyasu Ubayashi, Saeko Matsuura, and Seiichi Komiya. Association aspect. In Lieberherr [520], pages 16–25.

[735] F. Sánchez, J. Hernandez, J. M. Murillo, and E. Pedarza. Run-time adaptability of synchronization policies in concurrent object-oriented languages. In ECOOP-AOP98 [277].

[736] Cl/'audio Sant'Anna, Alessandro Garcia, Christina Chavez, Carlos Lucena, and Arndt Arndt von Staa. On the reuse and maintenance of aspect-oriented software: An assessment framework. In *XVII Brazilian Symposium on Software Engineering*, October 2003.

[737] Naomi Sapir, Shmuel Tyszberowicz, and Amiram Yehudai. Extending uml with aspect usage constraints in the analysis and design phases. In Kandé et al. [443].

[738] A. Sassen, G. Amoros, P. Donth, K. Geihs, J. Jézéquel, K. Odent, N. Plouzeau, and T. Weis. QCCS: A methodology for the development of contract-aware components based on aspect-oriented design. In AOSD-EA02 [48].

[739] Juha Savolainen. Improving product line development with subject-oriented programming. In ICSE-AOP00 [427].

[740] Juha Savolainen. Towards multi dimensional methods. In OOPSLA-AOP00 [628].

[741] Vibha Sazawal. Separation of concerns for ubiquitous computing. In ICSE-AOP01 [428].

[742] Nathanael Schärli and Franz Achermann. Partial evaluation of inter-language wrappers. In *Workshop on Composition Languages, WCL'01 (ESEC/FSE)*, September 2001.

[743] Arno Schmidmeier. Transferring persistence concepts in Java ODBMSs to AspectJ based on ODMG standards. In Rashid [706].

[744] Arno Schmidmeier. Hard times are good times for aosd. In Bodkin et al. [120].

[745] Arno Schmidmeier, Stefan Hanenberg, and Rainer Unland. Known concepts implemented in AspectJ. In Bachmendo et al. [62].

[746] Phillip Schmidt, Sergio Alvarado, Jaime Milstein, Gregory Mulert, Robert Duvall, and Jesus Rivera. A systems engineering perspective of aspect-oriented software architectural analysis using UML. In Aldawud et al. [30].

[747] R. Schmidt and U. Aßmann. Extending aspect-oriented programming in order to flexibly support workflows. In ICSE-AOP98 [429].

[748] Andreas I. Schmied and Franz J. Hauck. Composing non-orthogonal aspects. In Martn et al. [556].

[749] S. Schonger, E. Pulvermueller, and S. Sarstedt. Aspect oriented programming and component weaving: Using XML representations of abstract syntax trees. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[750] Wolfgang Schult and Andreas Polze. Speed vs. memory usage - an approach to deal with contrary aspects. In Coady et al. [200].

[751] Karsten Schulz and Maria E. Orlowska. Architectural issues for cross-organizational B2B interactions. In Choukair [165], pages 79–87.

[752] Mario Schüpany, Christa Schwanninger, and Egon Wuchner. Aspect-oriented programming for .NET. In AOSD-PAT02 [50].

[753] Christa Schwanninger and Egon Wuchner. Commercialization. In Bodkin et al. [120].

[754] Marc Segura-Devillechaise, Jean-Marc Menaud, Gilles Muller, and Julia L. Lawall. Web cache prefetching as an aspect: Towards a dynamic-weaving based solution. In Akşit [22], pages 110–119.

[755] Lionel Seinturier. JST: An object synchronization aspect for Java. In ECOOP-AOP99 [278].

[756] Linda Seiter. *Design Patterns for Managing Evolution*. PhD thesis, Northeastern University, 1996.

[757] Linda Seiter, Mira Mezini, and Karl Lieberherr. Dynamic component gluing. In Ulrich Eisenecker and Krzysztof Czarnecki, editors, *First Int'l Symposium on Generative and Component-Based Software Engineering*. Springer, 1999.

[758] Linda Seiter, Mira Mezini, and Karl Lieberherr. Dynamic component gluing. In OOPSLA-AOP99 [631].

[759] Linda M. Seiter, Jens Palsberg, and Karl J. Lieberherr. Evolution of Object Behavior using Context Relations. In David Garlan, editor, *Symposium on Foundations of Software Engineering, SIGSOFT*, pages 46–57, San Francisco, 1996. ACM Press (SIGSOFT).

[760] Linda M. Seiter, Jens Palsberg, and Karl J. Lieberherr. Evolution of object behavior using context relations. *IEEE Trans. Soft. Eng.*, 24(1):79–92, January 1998.

[761] Damien Sereni and Oege de Moor. Static analysis of aspects. In Akşit [22], pages 30–39.

[762] Viren Shah. 'Commercializing AOP: Open-sourcing the stable doors' and 'Software security: Aosd as the savior'. In Bodkin et al. [120].

[763] Mati Shomrat and Amiram Yehudai. Obvious or not? Regulating architectural decisions using aspect-oriented programming. In Kiczales [473], pages 3–9.

[764] Marcelo Sihman and Shmuel Katz. A calculus of superimpositions for distributed systems. In Kiczales [473], pages 28–40.

[765] Marcelo Sihman and Shmuel Katz. Model checking applications of aspects and superimpositions. In Leavens and Clifton [514].

[766] Marcelo Sihman and Shmuel Katz. Superimpositions and aspect-oriented programming. *The Computer Journal*, 46(5):529–541, September 2003.

[767] Antonio Rito Silva. Separation and composition of overlapping and interacting concerns. In OOPSLA-AOP99 [631].

[768] Ignacio Silva-Lepe. Abstracting graph-based specifications of object-oriented programs. In *ACM Computer Science Conference, Symposium on Applied Computing*, pages 447–451, Phoenix, Arizona, 1994. ACM.

[769] D. Simmonds, S. Ghosh, and R. France. An aspect-oriented model driven architectural framework for middleware transparency. In Araújo et al. [54].

[770] Ian Simmonds. Clues in the search for ever more valuable separations of concern. In ICSE-AOP00 [427].

[771] Ian Simmonds and David Ing. Clues in the search for ever more valuable separations of concern. In OOPSLA-AOP00 [628].

[772] Henny B. Sipma. A formal model for cross-cutting modular transition systems. In Leavens and Clifton [514].

[773] Mark Skipper. The Watson subject compiler and AspectJ (A critique of practical objects). In OOPSLA-AOP99 [631].

[774] Mark Skipper. A model of composition oriented programming. In ICSE-AOP00 [427].

[775] Mark Skipper. Semantics of an object-oriented language with aspects and advice. In ECOOP-AOP01 [275].

[776] Therapon Skotiniotis, Karl Lieberherr, and David Lorenz. Aspect instances and their interactions. In Bergmans et al. [98].

[777] Pawel Slowikowski and Krzysztof Zieli nski. Comparison study of aspect-oriented and container managed security. In Hannemann et al. [377].

[778] Yannis Smaragdakis and Don Batory. Mixin layers: An object-oriented implementation

technique for refinements and collaboration-based designs. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, 11(2):215–255, 2002.

[779] Rik Smoody. Aspects can be objects, too. In OOPSLA-AOP02 [630].

[780] Gregor Snelting and Frank Tip. Theory and formal techniques semantics-based composition of class hierarchies. In Magnusson [552], pages 562–584.

[781] S. Soares and P. Borba. PaDA: A pattern for distribution aspects. In *Second Latin American Conference on Pattern Languages of Programming — SugarLoafPLoP*, August 2002.

[782] Sérgio Soares and Paulo Borba. Concurrency control with java and relational databases. In *Proceedins of V Brazilian Symposium on Programming Languages, SBLP 2001*, pages 252–267, may 2001. (in Portuguese).

[783] Sérgio Soares and Paulo Borba. Concurrency manager. In *First Latin American Conference on Pattern Languages Programming, SugarLoafPLoP 2001*, oct 2001. Published in UERJ Magazine: Special Issue on Software Patterns, June 2002, pages 221-231.

[784] Sérgio Soares and Paulo Borba. Aspectj - aspect-oriented programming in java. In *Tutorial in Proceedins of VI Brazilian Symposium on Programming Languages, SBLP 2002*, pages 39–55, jun 2002. (in portuguese).

[785] Sérgio Soares and Paulo Borba. Concurrency control with java and relational databases. In *Proceeding of 26th Annual IEEE Computer Software and Applications Conference, COMPSAC 2002*, pages 843–849, aug 2002.

[786] Sérgio Soares and Paulo Borba. Pip: Progressive implementation pattern. In *1st Workshop on Software Development Process Patterns (OOPSLA02)*, nov 2002.

[787] Sergio Soares, Eduardo Laureano, and Paulo Borba. Implementing distribution and persistence aspects with AspectJ. In *Proceedings of the 17th ACM conference on Object-oriented programming, systems, languages, and applications*, pages 174–190. ACM Press, 2002.

[788] A. Speck, M. Clauss, and B. Franczyk. Concerns of variability in bottom-up product-lines. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[789] Andreas Speck, Elke Pulvermüller, and Mira Mezini. Reusability of concerns. In ECOOP-AOP00 [274].

[790] Olaf Spinczyk, Andreas Gal, and Wolfgang Schrder-Preikschat. AspectC++: an aspect-oriented extension to the C++ programming language. In *Proceedings of the Fortieth International Conference on Tools Pacific*, pages 53–60. Australian Computer Society, Inc., 2002.

[791] Erlend Stav. Extending a visual tool. A challenging problem for extending object-orientation. In ECOOP-AOP00 [274].

[792] Dominik Stein, Stefan Hanenberg, and Rainer Unland. Designing aspect-oriented crosscutting in UML. In AOSD-UML02 [52].

[793] Dominik Stein, Stefan Hanenberg, and Rainer Unland. On representing join points in the uml. In Kandé et al. [443].

[794] Dominik Stein, Stefan Hanenberg, and Rainer Unland. An UML-based aspect-oriented design notation. In Kiczales [473], pages 106–112.

[795] Dominik Stein, Stefan Hanenberg, and Rainer Unland. Position paper on aspect-oriented modeling: Issues on representing crosscutting features. In Aldawud et al. [30].

[796] Peter Sties and Wolfgang Kellerer. A generic and implementation independent service description model. In Choukair [165], pages 163–168.

[797] Zoran Stojanovic and Ajantha Dahanayak. Components and viewpoints as integrated separations of concerns in system designing. In AOSD-AOD02 [47].

[798] Maximilian Störzer. Analysis of AspectJ programs. In Bachmendo et al. [62].

[799] Maximilian Störzer and Jens Krinke. Interference analysis for AspectJ. In Leavens and Clifton [514].

[800] Maximilian Storzer, Jens Krinke, and Silvia Breu. Trace analysis for aspect application. In Hannemann et al. [377].

[801] G. T. Sullivan. Aspect-oriented programming using reflection and meta-object protocols. *Comm. ACM*, 44(10):95–97, October 2001.

[802] Gregory T. Sullivan. Aspect-oriented programming using reflection. In OOPSLA-AOP01 [629].

[803] Kevin Sullivan, Lin Gu, and Yuanfang Cai. Non-modularity in aspect-oriented languages: Integration as a crosscutting concern for AspectJ. In Kiczales [473], pages 19–27.

[804] Thanwadee Sunetnanta and Anthony Finkelstein. Automated consistency checking for multiperspective software specifications. In ICSE-AOP01 [428].

[805] S. Sutton Jr. Concerns in a requirements model - a small case study. In Araújo et al. [54].

[806] S. M. Sutton Jr. Early-stage concern modeling. In AOSD-EA02 [48].

[807] Stanley Sutton Jr. and Isabelle Rouvellou. Modeling of software concerns in Cosmos. In Kiczales [473], pages 127–133.

[808] Stanley M. Sutton Jr. Multiple dimensions of concern in software testing. In OOPSLA-AOP99 [631].

[809] Stanley M. Sutton Jr. and Isabelle Rouvellou. Concerns in the design of a software cache. In OOPSLA-AOP00 [628].

[810] Stanley M. Sutton Jr. and Isabelle Rouvellou. Applicability of categorization theory to multidimensional separation of concerns. In OOPSLA-AOP01 [629].

[811] Stanley M Sutton, Jr. and Isabelle Rouvellou. Issues in the design and implementation of a concern-space modeling schema. In ICSE-AOP01 [428].

[812] Stanley M. Sutton Jr. and Peri Tarr. Aspect-oriented design needs concern modeling. In AOSD-AOD02 [47].

[813] Davy Suvée and Wim Vanderperren. JAsCo: an aspect-oriented approach tailored for component based software development. In Akşit [22], pages 21–29.

[814] Junichi Suzuki and Yoshikazu Yamamoto. Extending UML with aspects: Aspect support in the design phase. In ECOOP-AOP99 [278].

[815] E. Tanter and P. Ebraert. A flexible approach to interactive runtime inspection. In Filman et al. [296].

[816] Peri Tarr, William Harrison, Harold Ossher, Anthony Finkelstein, Bashar Nuseibeh, and Dewayne Perry. Workshop on multi-dimensional separation of concerns in software engineering. *ACM SIGSOFT Software Engineering Notes*, 26(1):78–81, 2001.

[817] Peri Tarr, William Harrison, Harold Ossher, Anthony Finkelstein, Bashar Nuseibeh, and Dewayne Perry. Workshop summary. In ICSE-AOP00 [427].

[818] Peri Tarr, Harold Ossher, William Harrison, and Stanley M. Sutton Jr. N degrees of separation: Multi-dimensional separation of concerns. In *Proc. 21st Int'l Conf. Software Engineering (ICSE'1999)*, pages 107 – 119. IEEE Computer Society Press, May 1999.

[819] Peri Tarr, Harold Ossher, and Johannes Henkel. Visualization as an aid to compositional software engineering. In OOPSLA-AOP01 [629].

[820] M. Tatsubori. Separation of distribution concerns in distributed Java programming. In *OOPSLA'01, Doctoral Symposium*, October 2001.

[821] B. Tekinerdogan. ASAAM: Aspectual software architecture analysis method. In Araújo et al. [54].

[822] Bedir Tekinerdogan, Paul Clements, Ana Moreira, and Jo ao Araújo, editors. *Early Aspects: Aspect-Oriented Requirements Engineering and Architecture Design*, March 2004.

[823] B. Tekinerdoğan and M. Akşit. Adaptability in object-oriented development. In M. Muhlhauser, editor, *Special Issues in Object-Oriented Programming*, pages 7–11, 1996.

[824] B. Tekinerdoğan and M. Akşit. Deriving design aspects from canonical models. In ECOOP-AOP98 [277].

[825] Bedir Tekinerdoğan and Mehmet Akşit. Separation and composition of concerns through synthesis-based design. In OOPSLA-AOP00 [628].

[826] Aleksandra Tesanovic, Jörgen Hansson, Dag Nyström, and Christer Norström. Integrating symbolic worst-case execution time analysis with aspect-oriented system development. In OOPSLA-AOP02 [630].

[827] Craig Thompson, Paul Pazandak, Venu Vasudevan, Frank Manola, Mark Palmer, Gil Hansen, and Tom Bannon. Intermediary architecture: Interposing middleware object services between web client and server. *ACM Computing Surveys*, 31(2es):14, 1999.

[828] Kresten Krab Thorup. Contextual class extensions. In ECOOP-AOP97 [276].

[829] M. Tokoro and R. Pareschi, editors. *Proc. 8th European Conf. Object-Oriented Programming*. Springer Verlag LNCS 821, July 1994.

[830] Soren Top, Bo Nørregaard, and Christo Angelov. Separation of fault tolerant control concern in embedded control system. In AOSD-AOD02 [47].

[831] Tom Tourwé, Johan Brichau, and Kris Gybels. On the existence of the aosd-evolution paradox. In Bergmans et al. [98].

[832] E. Truyen, B. Vanhaute, W. Joosen, P. Verbaeten, and B. Nørregaard Jørgensen. A dynamic customization model for distributed component-based systems. In Choukair [165], pages 147–156.

[833] Eddy Truyen, Wouter Joosen, and Pierre Verbaeten. Run-time support for aspects in distributed system infrastructure. In AOSD-PAT02 [50].

[834] Eddy Truyen, Bo Nøorregaard Jørgensen, Wouter Joosen, and Pierre Verbaeten. Aspects for run-time component integration. In ECOOP-AOP00 [274].

[835] Eddy Truyen, Bo Nørregaard Jørgensen, and Wouter Joosen. Customization of object request brokers through dynamic reconfiguration. In *Proc. of Tools Europe 2000*, June 2000.

[836] Eddy Truyen, Bart Vanhaute, Wouter Joosen, Pierre Verbaeten, and Bo Nørregaard Jørgensen. Customization of on-line services with simultaneous client-specific views. In ECOOP-AOP01 [275].

[837] Eddy Truyen, Bart Vanhaute, Wouter Joosen, Pierre Verbaeten, and Bo Nørregaard Jørgensen. Customization of on-line services with simultaneous client-specific views. In ICSE-AOP01 [428].

[838] Eddy Truyen, Bart Vanhaute, Wouter Joosen, Pierre Verbaeten, and Bo Nørregaard Jørgensen. Dynamic and selective combination of extensions in component-based applications. In *Proc. 23rd Int'l Conf. Software Engineering (ICSE'2001)*, May 2001.

[839] David B. Tucker and Shriram Krishnamurthi. Pointcuts and advice in higher-order languages. In Akşit [22], pages 158–167.

[840] Naoyasu Ubayashi and Tetsuo Tamai. Separation of concerns in mobile agent applications. In Yonezawa and Matsuoka [885], pages 89–109.

[841] Naoyasu Ubayashi and Tetsuo Tamai. Aspect oriented programming with model checking. In Kiczales [473], pages 148–154.

[842] David Ungar. The limits to factoring. In OOPSLA-AOP99 [631].

[843] Peter Van Roy, Seif Haridi, Per Brand, Gert Smolka, Michael Mehl, and Ralf Scheidhauer. Using mobility to make transparent distribution practical. In ECOOP-AOP97 [276].

[844] Koenraad Vandenborre, Muna Matar, and Ghislain Hoffman. Orthogonal persistence using aspect oriented programming. In AOSD-PAT02 [50].

[845] Glenn Vanderburg. Position paper. In *OOPSLA 2001 Software Archaeology Workshop*, October 2001.

[846] Wim Vanderperren. A pattern based approach to separate tangled concerns in component based development. In AOSD-PAT02 [50].

[847] Wim Vanderperren, Davy Suvée, and Viviane Jonckers. Invasive composition adapters: An aspect-oriented approach for visual component-based development. In Coady et al. [200].

[848] Bart Vanhaute, Bart De Win, and Bart De Decker. Building frameworks in AspectJ. In ECOOP-AOP01 [275].

[849] Bart Vanhaute, Eddy Truyen, Wouter Joosen, and Pierre Verbaeten. Composing non-orthogonal meta-programs. In OOPSLA-AOP99 [631].

[850] Matthias Veit and Stephan Herrmann. Model-view-controller and object teams: A perfect match of paradigms. In Akşit [22], pages 140–149.

[851] John Viega. Separation of concerns for security. In ICSE-AOP00 [427].

[852] D. Vollmann. Visibility of join-points in AOP and implementation languages. In Costanza et al. [229]. Technical report IAI-TR-2002-1, ISSN 0944-8535.

[853] Christina von Flach G. Chavez and Carlos J. P. de Lucena. Design-level support for aspect-oriented software development. In OOPSLA-AOP01 [629].

[854] Valentino Vranić. AspectJ paradigm model: A basis for multi-paradigm design for AspectJ. In Bosch [125], pages 48–57.

[855] D. Wagelaar. A concept-based approach for early aspect modelling. In Araújo et al. [54].

[856] Dennis Wagelaar and Lodewijk Bergmans. Using a concept-based approach to aspect-oriented software design. In AOSD-AOD02 [47].

[857] David Walker, Steve Zdancewic, and Jay Ligatti. A theory of aspects. In *Proceedings of the ACM SIGPLAN International Conference on Functional Programming*, August 2003.

[858] R. J. Walker, E. L. A. Baniassad, and G. Murphy. Assessing aspect-oriented programming and design. In ECOOP-AOP98 [277].

[859] Robert J. Walker. Supporting inconsistent world views. In Bergmans et al. [98].

[860] Robert J. Walker, Elisa L. A. Baniassad, and Gail C. Murphy. An initial assessment of aspect-oriented programming. In *Proc. 21st Int'l Conf. Software Engineering (ICSE '99)*, pages 120–130, 1999.

[861] Robert J. Walker and Gail C. Murphy. Joinpoints as ordered events: Towards applying implicit context to aspect-orientation. In ICSE-AOP01 [428].

[862] Mitchell Wand. Understanding aspects: extended abstract. In *Proceedings of the Eighth ACM SIGPLAN International Conference on Functional Programming*, pages 299–300. ACM Press, 2003.

[863] Mitchell Wand, Gregor Kiczales, and Chris Dutchyn. A semantics for advice and dynamic join points in aspect-oriented programming. In AOSD-FOAL02 [49], pages 1–8.

[864] Hans Wegener and Ahmed Rida. Reengineering of metalevel abstractions with data mining methods. In OOPSLA-AOP00 [628].

[865] Ian Welch and Robert Stroud. Load-time application of aspects to Java COTS software. In ECOOP-AOP99 [278].

[866] Ian Welch, Robert J. Stroud, and Alexander Romanovsky. Aspects of exceptions at the meta-level. In Yonezawa and Matsuoka [885], pages 280–282.

[867] Ian S. Welch and Robert J. Stroud. Security and aspects: A metaobject protocol viewpoint. In AOSD-PAT02 [50].

[868] Ian S. Welch and Robert J. Stroud. Reengineering security as a crosscutting concern. *The Computer Journal*, 46(5):578–589, September 2003.

[869] Ian S. Welch, Robert J. Stroud, and Alexander Romanovsky. Aspects of exceptions at the meta-level. In Rashid [706].

[870] M. Wermelinger, J. L. Fiadeiro, L. Andrade, G. Koutsoukos, and J. Gouveia. Separation of core concerns: Computation, coordination, and configuration. In OOPSLA-AOP01 [629].

[871] Jon Whittle, Jo ao Araújo, and Dae-Kyoo Kim. Modeling and validating interaction aspects in UML. In Aldawud et al. [31].

[872] J. C. Wichman. The development of a preprocessor to facilitate composition filters in the Java language. Master's thesis, University of Twente, 1999.

[873] Edward Willink and Vyacheslav Muchnick. Weaving a way past the C++ one definition rule. In ECOOP-AOP99 [278].

[874] B. De Win, J. Van den Bergh, F. Matthijs, B. De Decker, and W. Joosen. A security architecture for electronic commerce applications. In S. Qing and J. Eloff, editors,

Information Security for Global Information Infrastructures, pages 491–500. Kluwer Academic Publishers, 2000.

[875] Bart De Win, Wouter Joosen, and Frank Piessens. AOSD & security: A practical assessment. In Bergmans et al. [98].

[876] Bart De Win, Viren Shah, Wouter Joosen, and Ron Bodkin, editors. *AOSDSEC: AOSD Technology for Application-Level Security*, March 2004.

[877] Bart De Win, Bart Vanhaute, and Bart De Decker. How aspect-oriented programming can help to build secure software. *Informatica*, 26(2):141–149, 2001.

[878] Jan Wloka. Refactoring in the presence of aspects. In Martn et al. [556].

[879] Eric Wohlstadter and Prem Devanbu. A lazy approach to separating architectural concerns. In ICSE-AOP01 [428].

[880] Eric Wohlstadter, Aaron Keen, Stoney Jackson, and Premkumar Devanbu. Accommodating evolution in AspectJ. In OOPSLA-AOP01 [629].

[881] Eric Wohlstadter, Brian Toone, and Prem Devanbu. A framework for flexible evolution in distributed heterogeneous systems. In *Proceedings of the Workshop on Principles of Software Evolution*, pages 39–42. ACM Press, 2002.

[882] Pengcheng Wu, Shriram Krishnamurthi, and Karl Lieberherr. Traversing recursive object structures: The functional visitor in demeter. In Bergmans et al. [98].

[883] Cun Xiao. *Adaptive Software: Automatic Navigation Through Partially Specified Data Structures*. PhD thesis, Northeastern University, 1994.

[884] Annie T. T. Ying, Gail C. Murphy, Raymond T. Ng, and Mark C. Chu-Carroll. Using version information for concern interface and code-assist. In OOPSLA-AOP02 [630].

[885] A. Yonezawa and S. Matsuoka, editors. *Metalevel Architectures and Separation of Crosscutting Concerns 3rd Int'l Conf. (Reflection 2001), LNCS 2192*. Springer-Verlag, September 2001.

[886] A. Zaidman and S. Demeyer. Using a variant of sliding window to reduce event trace data. In Filman et al. [296].

[887] Aida Atef Zakaria and Hoda Hosny. Metrics for aspect-oriented software design. In Aldawud et al. [30].

[888] Aida Atef Zakaria, Hoda Hosny, and Amir Zeid. A uml extension for modeling aspect-oriented systems. In Kandé et al. [443].

[889] Uwe Zdun. Using architecture recovery and tracing patterns for aspect composition. In Bachmendo et al. [62].

[890] Charles Zhang and H-A. Jacobsen. Quantifying aspects in middleware platforms. In Akşit [22], pages 130–139.

[891] Charles Zhang and Hans-Arno Jacobsen. Tinyc^2: Towards building a dynamic weaving aspect language for C. In Leavens and Clifton [514].

[892] Hongyu Zhang, Stan Jarzabek, and Soe Myat Swe. X-frames approach for handling variants within concerns. In ICSE-AOP01 [428].

[893] Hongyu Zhang, Stan Jarzabek, and Soe Myat Swe. XVCL approach to separating concerns in product family assets. In Bosch [125], pages 36–47.

[894] Yan Zhang, Anna Liu, and Wei Qu. Implementing performance "tactics" using aspect-oriented programming. In Bachmendo et al. [62].

[895] J. Zhao. Towards a metrics suite for aspect-oriented software. Technical Report SE-136-25, Information Processing Society of Japan (IPSJ), March 2002.

[896] J. Zhao. Unit testing for aspect-oriented programs. Technical Report SE-141-6, Information Processing Society of Japan (IPSJ), May 2003.

[897] J. Zhao and M. Rinard. Pipa: A behavioral interface specification language for aspectj. In *Proc. Fundamental Approaches to Software Engineering (FASE'2003), LNCS 2621*, pages 150–165. Springer-Verlag, April 2003.

[898] J. Zhao and M. Rinard. System dependence graph construction for aspect-oriented programs. Technical Report MIT-LCS-TR-891, Laboratory for Computer Science, MIT, March 2003.

[899] Jianjun Zhao. Change impact analysis for aspect-oriented software evolution. In *Proceedings of the Workshop on Principles of Software Evolution*, pages 108–112. ACM Press, 2002.

[900] Jianjun Zhao. Slicing aspect-oriented software. In *Proc. 10th IEEE International Workshop on Program Comprehension*, pages 251–260, June 2002.

[901] Jianjun Zhao. Tool support for unit testing of aspect-oriented software. In OOPSLA-AOP02 [630].

[902] Jianjun Zhao. Data-flow-based unit testing of aspect-oriented programs. In *27th Annual International Computer Software and Applications Conference (COMPSAC)*, pages 188–197. IEEE Computer Society, November 2003.

[903] John Zinky, Joe Loyall, Partha Pal, Richard Shapiro, Richard Schantz, James Megquier, Michael Atighetchi, Craig Rodrigues, and David Karr. An AOP challange problem: Managing QoS on interactions between distributed objects. In ECOOP-AOP00 [274].

[904] John Zinky and Richard Shapiro. The aspect-oriented interceptors' pattern for crosscutting and separation of concerns using conventional object oriented programming languages. In Coady et al. [200].

[905] John Zinky, Richard Shapiro, Joe Loyall, Partha Pal, and Michael Atighetchi. Separation of concerns for reuse of systemic adaptation in QuO 3.0. In ECOOP-AOP01 [275].

[906] John A. Zinky, David E. Bakken, and Richard D. Schantz. Architectural support for quality of service for CORBA objects. *Theory and Practice of Object Systems*, 3(1):55–73, January 1997.